



**SENSOR ULTRASONIK SEBAGAI ALAT NAVIGASI ROBOT  
PADA ROBOT PEMADAM API BERBASIS  
MIKROKONTROLER ATMega 8535**

**TUGAS AKHIR**

**Disusun Guna Memenuhi Salah Satu Syarat Untuk Menyelesaikan Studi di  
Program Studi Diploma III Teknik Elektro Fakultas Teknik  
Universitas Diponegoro**

**Disusun Oleh :  
DELTA AGUS SETYA ABADI  
L0F 005 464**

**PROGRAM STUDI DIPLOMA III TEKNIK ELEKTRO  
F A K U L T A S T E K N I K  
UNIVERSITAS DIPONEGORO  
SEMARANG  
2008**

## DAFTAR ISI

BAB I	PENDAHULUAN	
1.1	Latar Belakang .....	1
1.2	Tujuan .....	2
1.3	Pembatasan Masalah .....	3
1.4	Metode Penulisan .....	3
1.5	Sistematika Penulisan .....	4
BAB II	LANDASAN TEORI	
2.1	Robot Avoider .....	6
2.2	Sensor Ultrasonik .....	18
2.2.1	Sensor Jarak Ultrasonik Ping .....	9
2.2.2	Sensor Jarak Ultrasonik SRF 04 .....	12
2.2.3	Karakteristik sensor Ultrasonik.....	13
2.2.4	Prinsip Dasar Navigasi Robot.....	15
2.3	Mikrokontroler ATmega 8535 .....	17
2.3.1	Susunan kaki Mikrokontroler ATmega 8535 .....	18
2.3.2	Blok diagram dan Arsitektur ATmega 8535 .....	19
2.3.3	Organisasi memori .....	20
2.3.4	Memori Program .....	21
2.3.5	Memori Data .....	22
2.3.6	Register Serba Guna .....	23
2.3.7	Pewaktu CPU .....	24
2.3.8	PORT A/B/C/D/E, DDR A/B/C/D/E, DAN PIN A/B/C/D/E.....	26

2.3.9	Pewaktu atau Pencacah .....	26
2.4	Program Pendukung .....	28
2.4.1	Code Vision AVR Full Version 1.24.2 Series .....	28
2.4.2	ISP (In-System Programming).....	30
2.5	Catu Daya.....	31
 <b>BAB III PRINSIP KERJA RANGKAIAN</b>		
3.1	Cara Kerja Tiap Blok .....	34
3.1.1	Cara Kerja Sensor Ultrasonik Sebagai Navigasi Robot .....	34
3.1.2	Cara Kerja Mikrokontroler ATmega 8535 .....	37
3.1.3	Program Code Vision AVR.....	39
3.1.4	Prinsip Kerja Catu Daya .....	40
3.2	Cara kerja keseluruhan.....	42
 <b>BAB IV PEMBUATAN BENDA KERJA</b>		
4.1	Alat dan bahan yang dibutuhkan.....	47
4.1.1	Daftar alat.....	47
4.1.2	Daftar bahan .....	48
4.2	Pembuatan bagian elektronika .....	49
4.2.1	Pembuatan PCB .....	49
4.2.1.2	Proses pembuatan jalur prt.....	49
4.2.1.3	Proses pelarutan PCB .....	50
4.2.2	Pengeboran PCB .....	51
4.2.3	Pemasangan komponen.....	51
4.3	Pembuatan bagian mekanik.....	52
4.3.1	Pembuatan body robot .....	52
4.3.2	Penempelan skotlet.....	53
4.3.3	Pengisian program.....	53
4.4	Proses perakitan .....	54

<b>BAB V</b>	<b>PENGUJIAN DAN PENGUKURAN</b>	
5.1	Tujuan .....	58
5.2	Peralatan yang digunakan .....	58
5.3	Langkah – langkah pengujian .....	59
5.4	Pengujian rangkaian catu daya.....	59
5.5	Pengujian sistim minimum AVR .....	60
5.6	Pengujian sensor ping .....	62
5.7	Analisa hasil pengujian .....	64
5.8	Analisa rangkaian mikrokontroler.....	64
5.9	Analisa sensor ping .....	64
5.10	Spesifikasi benda kerja.....	65
 <b>BAB VI</b>	 <b>PENUTUP</b>	
6.1	Kesimpulan .....	66
6.2	Saran.....	67

## DAFTAR PUSTAKA

## LAMPIRAN-LAMPIRAN

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan teknologi sekarang ini telah mengalami peningkatan sedemikian pesatnya hingga ke berbagai sisi kehidupan manusia. Perkembangan tersebut didukung oleh tersedianya perangkat keras (hardware) maupun perangkat lunak (software) yang semakin canggih dan meningkat kemampuannya. Perpaduan antara hardware dan software bisa membuat suatu sistem yang canggih yang tentunya lebih menghemat dari segi biaya pembuatan maupun biaya pemeliharaan. Apabila terjadi kerusakan, kerusakan tersebut tidak hanya berasal dari hardwarenya namun bisa berasal dari softwarenya. Sistem dapat diperbaiki melalui softwarenya dengan cara menghapus dan isi dengan software yang baru tanpa mengeluarkan biaya lagi.

Munculnya sensor – sensor yang semakin canggih dunia elektronika saat ini dapat menunjang manusia untuk membuat software. Manusia diharapkan dapat membuat software yang bisa digunakan untuk menunjang hardware–hardware untuk menjadi suatu sistem yang sangat canggih dan tentunya akan sangat berguna untuk mempermudah pekerjaan manusia. Berdasarkan hal tersebut di atas, penulis mencoba untuk merancang sebuah robot untuk memadamkan api.dengan perpaduan antara hardware ( sensor-sensor ) dan software. Hal ini difungsikan agar robot ini dapat berjalan memadamkan api sesuai dengan yang penulis harapkan. Dalam proses pemadaman api tersebut robot berjalan dengan cara meniti dinding menggunakan sensor ultrasonik pada robot sehingga robot

tersebut terus berjalan meniti dinding tanpa harus menaberak dinding, jadi robot dapat terus berjalan mencari nyala api untuk dipadamkan.

Sensor yang bekerja sebagai navigasi robot adalah sensor ultrasonik dimana sensor ini mampu mendeteksi adanya objek bekisar antara 3 cm – 3 m, jarak yang dideteksi sensor menjadi acuan bagi robot untuk menentukan arah belokan, sehingga robot tidak dapat meneberak halangan dan dapat terus berjalan mencari nyala api untuk dipadamkan. hal ini yang dijadikan bagi penulis untuk menentukan judul tugas akhir, oleh karena itu dibuatlah tugas akhir dengan judul **“SENSOR ULTRASONIC SEBAGAI ALAT NAVIGASI ROBOT PADA ROBOT PEMADAM API BERBASIS MIKROKONTROLER ATMega 8535”**

## **1.2 Tujuan**

Tujuan dari pembuatan tugas akhir ini adalah :

1. Memenuhi salah satu syarat untuk menyelesaikan pendidikan pada Program Studi Diploma III Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang.
2. Mengaplikasikan ilmu yang telah dipelajari dan diperoleh selama menempuh pendidikan pada Program Studi Diploma III Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang.
3. Mengaplikasikan suatu mikrokontroller yang akan mengontrol sensor ultrasonik sehingga dapat menjalankan robot dengan cara meniti dinding untuk bisa menemukan nyala api.

4. Mengetahui lebih dalam mengenai penggunaan mikrokontroller ATmega 8535 yang digunakan sebagai pengendali sistem dari segi hardware maupun software.

### **1.3 Pembatasan Masalah**

Dalam pembuatan Tugas Akhir ini penulis memberikan pokok pembahasan yang mencakup beberapa hal, diantaranya sebagai berikut :

- 1 Prinsip kerja Mikrokontroller ATmega 8535 sebagai otak dari robot pemadam api
2. Pemrograman Mikrokontroller ATmega 8535 dengan bahasa pemrograman bahasa C untuk dapat menjalankan robot menggunakan code vision AVR
3. Sensor ultrasonik sebagai alat navigasi pada robot pemadam api.

### **1.4 Metode Penulisan**

Dalam perencanaan dan pembuatan alat ini, penulis menggunakan metode sebagai berikut :

1. Studi Kepustakaan

Metode ini dilakukan dengan cara melihat dan mencari literatur untuk memperoleh data yang berhubungan dengan alat yang dibuat.

2. Metode Observasi

Metode ini merupakan metode dengan melakukan penelitian dan mempelajari peralatan yang sudah ada untuk dikembangkan penulis menjadi suatu sistem yang dapat bekerja dengan baik, salah satunya penulis melakukan observasi pada kontes robot cerdas ( KRCI ) region II di UGM

yogyakarta 2008 guna mendapatkan pandangan tentang robot seperti yang telah direncanakan dalam konsep tugas akhir penulis

### 3. Metode Eksperimen

Metode ini dilakukan dengan melakukan serangkaian kegiatan di dalam ruang kerja ( bengkel ) yang meliputi perancangan, uji-coba, pembuatan, pengukuran, pengujian dan analisa benda kerja.

### 4. Metode Analisis Data

Metode analisis data yang dipakai adalah membandingkan data hasil dari pengukuran pada pembacaan masing – masing sensor dengan ketentuan sesuai dengan teori yang ada Metode ini juga menganalisa sistem kerja rangkaian secara keseluruhan sebagaimana yang diharapkan.

## **1.5 Sistematika Penulisan**

Untuk memberi gambaran yang jelas tentang susunan materi yang dibahas dalam Laporan Tugas Akhir ini, sistematika yang digunakan adalah sebagai berikut :

### **BAB I PENDAHULUAN**

Bab ini membahas mengenai Latar Belakang, Tujuan, Pembatasan Masalah, Metode Penulisan dan Sistematika Penulisan

### **BAB II LANDASAN TEORI**

Pada bab ini akan dibahas mengenai dasar teori dari masing–masing bagian yang menjadi paduan atau dasar dari pembuatan tugas akhir,



diantaranya sensor ultrasonik ping dari parallax, prinsip dasar navigasi robot, mikrokontroler ATmega 8535, catu daya dan code vision AVR C compiler

### **BAB III PRINSIP KERJA RANGKAIAN**

Pada bab ini akan menerangkan mengenai cara kerja sensor ultrasonik sebagai alat navigasi pada robot pemadam api, catu daya, mikrokontroler ATmega 8535

### **BAB IV PEMBUATAN BENDA KERJA**

Pada bab ini membahas mengenai proses perancangan dan pembuatan benda kerja baik perangkat lunak maupun perangkat keras sistem serta bahan dan alat yang dipergunakan.

### **BAB V PENGUKURAN DAN PENGUJIAN RANGKAIAN**

Dalam bab ini akan membahas tentang uji coba rangkaian dan alat apakah rangkaian telah berjalan sesuai dengan yang direncanakan, hasil pengukuran, hasil pengujian dan analisa hasil pengujian dan pengukuran dengan teori-teori yang ada.

### **BAB VI PENUTUP**

Berisi kesimpulan dan saran yang berkaitan dengan seluruh proses perancangan dan pembuatan tugas akhir ini serta penyelesaian laporannya.

### **DAFTAR PUSTAKA**

### **LAMPIRAN – LAMPIRAN**

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Robot Avoider**

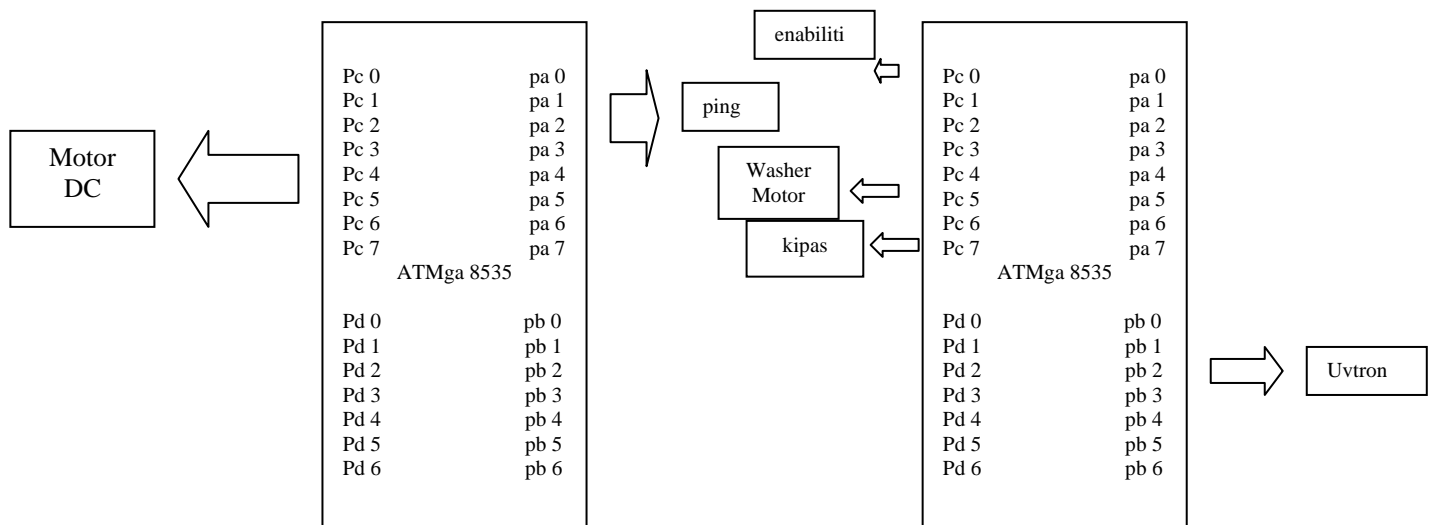
Robot avoider adalah robot beroda atau berkaki yang diprogram untuk dapat menghindari jika ada halangan, misalnya dinding. Robot avoider minimal membutuhkan tiga buah sensor untuk mendeteksi penghalang yaitu sensor depan, kanan dan kiri. Dalam hal ini sensor yang dipergunakan adalah sensor ultrasonik.

Robot membutuhkan sensor yang banyak untuk hasil pendeteksian penghalang yang lebih baik. Hal ini dikarenakan keterbatasan sudut pancaran sensor ( biasanya sekitar 15 derajat saja ). Sudut pantulan yang terlalu besar akan menyebabkan hasil pembacaan sensor yang kurang akurat. Robot avoider adalah cikal bakal dari robot pemadam api dimana konsep dari robot avoider ini dapat berubah menjadi robot pemadam api dengan menambahkan sensor UVtron ke dalam robot ini maka akan menjelma menjadi robot pemadam api. Untuk membuat robot pemadam api tidaklah sulit yaitu dengan menambahkan sensor UVtron kedalam robot avoider. Sistem minimal robot avoider adalah sebagai berikut :

- 1 Mikrokontroller ATmega 8535
2. 3 buah sensor pendeteksi penghalang ( ultrasonik )
3. 2 buah motor dc sebagai penggerak roda

Untuk menjadi sebuah robot yang cerdas maka konsep robot avoider tadi dapat diubah menjadi robot pemadam api, yaitu dengan memberikan tambahan sebuah sensor pendeteksi api yaitu flame detector ( Uvtron ).

## Blok diagram robot pemadam api



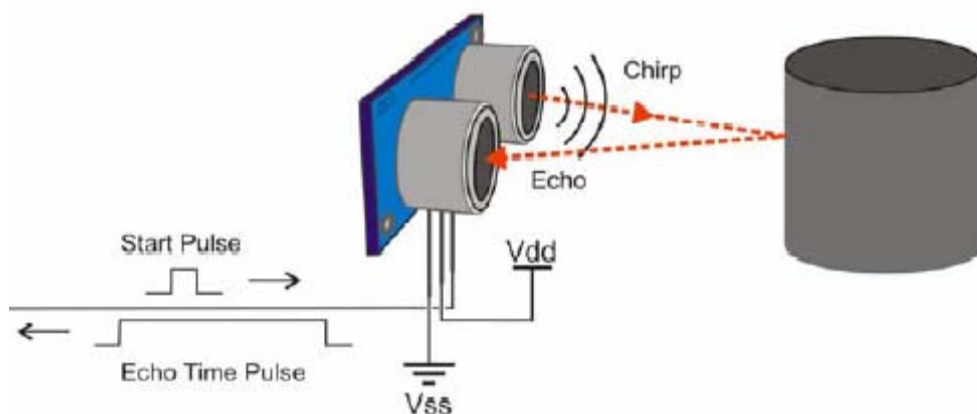
Gambar 2.1 Blok diagram robot pemadam api

## 2.2 Sensor Ultrasonik

Sensor ultrasonik adalah sensor yang bekerja berdasarkan prinsip pantulan gelombang suara dan digunakan untuk mendeteksi keberadaan suatu objek tertentu di depannya, frekuensi kerjanya pada daerah di atas gelombang suara dari 40 KHz hingga 400 KHz. Sensor ultrasonik terdiri dari dua unit, yaitu unit pemancar dan unit penerima. Struktur unit pemancar dan penerima sangatlah sederhana, sebuah kristal piezoelectric dihubungkan dengan mekanik jangkar dan hanya dihubungkan dengan diafragma penggetar. Tegangan bolak-balik yang memiliki frekuensi kerja 40 KHz – 400 KHz diberikan pada plat logam. Struktur atom dari kristal piezoelectric akan berkontraksi (mengikat), mengembang atau menyusut terhadap polaritas tegangan yang diberikan dan ini disebut dengan efek piezoelectric.

Kontraksi yang terjadi diteruskan ke diafragma penggetar sehingga terjadi gelombang ultrasonik yang dipancarkan ke udara (tempat sekitarnya). Pantulan

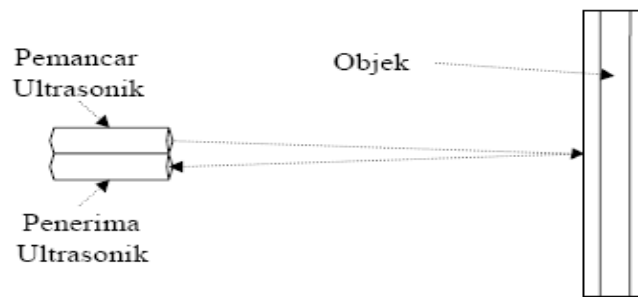
gelombang ultrasonik akan terjadi bila ada objek tertentu dan pantulan gelombang ultrasonik akan diterima kembali oleh unit sensor penerima. Selanjutnya unit sensor penerima akan menyebabkan diafragma penggetar akan bergetar dan efek piezoelectric menghasilkan sebuah tegangan bolak-balik dengan frekuensi yang sama. Untuk lebih jelas tentang prinsip kerja dari sensor ultrasonik dapat dilihat prinsip dari sensor ultrasonic pada gambar 2.2 berikut :



Gambar 2.2 Prinsip kerja sensor ultrasonik<sup>1</sup>

Besar amplitudo sinyal elektrik yang dihasilkan unit sensor penerima tergantung dari jauh dekatnya objek yang dideteksi serta kualitas dari sensor pemancar dan sensor penerima. Proses sensing yang dilakukan pada sensor ini menggunakan metode pantulan untuk menghitung jarak antara sensor dengan obyek sasaran. Jarak antara sensor tersebut dihitung dengan cara mengalikan setengah waktu yang digunakan oleh sinyal ultrasonik dalam perjalanannya dari rangkaian pengirim sampai diterima oleh rangkaian penerima, dengan kecepatan rambat dari sinyal ultrasonik tersebut pada media rambat yang digunakannya, yaitu udara. Prinsip pantulan dari sensor ultrasonik ini dapat dilihat pada gambar 2.3 sebagai berikut :

<sup>1</sup>[www.parallax.com](http://www.parallax.com),senin 16 juni 2008,12.36



Gambar 2.3 Prinsip pemantulan sensor ultrasonik<sup>2</sup>

Terdapat 2 jenis sensor ultrasonik yang beredar di pasaran yaitu :

- 1 Sensor ultrasonik ping ( parallax )
  - a. Sensor ultrasonik defantech ( SRF 04 ranger )

### 2.2.1 Sensor Jarak Ultrasonik Ping

Sensor jarak ultrasonik ping adalah sensor 40 khz produksi parallax yang banyak digunakan untuk aplikasi atau kontes robot cerdas. Kelebihan sensor ini adalah hanya membutuhkan 1 sinyal ( SIG ) selain jalur 5 v dan ground. Perhatikan gambar dibawah ini :



Gambar 2.4 Sensor jarak ultrasonik ping<sup>3</sup>

Sensor PING mendeteksi jarak objek dengan cara memancarkan gelombang ultrasonik ( 40 KHz ) selama  $t = 200$  us kemudian mendeteksi pantulannya. Sensor PING memancarkan gelombang ultrasonik sesuai dengan kontrol dari mikrokontroller pengendali ( pulsa trigger dengan  $t_{out}$  min 2 us )

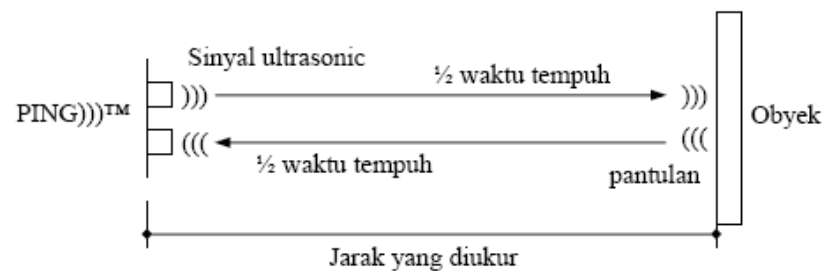
Spesifikasi sensor ultrasonik PING :

<sup>2</sup> Ibid

<sup>3</sup> Widodo Budiharjo, *Membuat Robot Cerdas*, 2007

1. Kisaran pengukuran 3 cm – 3 m
2. Input trigger – positive TTL pulse, 2 us min, 5 us tipikal
3. Echo hold off 750 us dari of trigger pulse
4. Delay before next measurement 200 us
5. Brust indikator LED menampilkan aktivitas sensor

Gelombang ini melalui udara dengan kecepatan 344 m/s kemudian mengenai obyek dan memantul kembali ke sensor. Ping mengeluarkan pulsa output high pada pin SIG setelah memancarkan gelombang ultrasonik dan setelah gelombang pantulan terdeteksi Ping akan membuat output low pada pin SIG. Lebar pulsa High ( $t_{IN}$ ) akan sesuai dengan lama waktu tempuh gelombang ultrasonik untuk 2x jarak ukur dengan obyek. Maka jarak yang diukur ialah  $[(t_{IN} \text{ s} \times 344 \text{ m/s}) : 2]$  meter.



Gambar 2.5 Jarak Ukur Sensor Ping<sup>4</sup>

Sistem minimal mikrokontroller ATmega 8535 dan software basic stamp Editor diperlukan untuk memprogram mikrokontroller dan mencoba sensor ini. Keluaran dari pin SIG ini yang dihubungkan ke salah satu port di kit mikrokontroller. Berikut contoh aplikasi sensor PING pada mikrokontroler BS2, dimana pin SIG terhubung ke pa pin7, dan memberikan catu daya 5V dan ground. fungsi **SIGOUT** untuk

<sup>4</sup> [www.parallax.com](http://www.parallax.com),senin 16 juni 2008,12.30

mentrigger ping, sedangkan fungsi **SIGIN** digunakan untuk mengukur pulsa yang sesuai dengan jarak dari objek target.

### Test ping dengan Code Vision AVR

```
/* program powered by delta_cakep progamer*/
#include <mega8535.h>
#include <stdio.h>
#include <delay.h>
#define PULSA PORTB.0
#define echo PINB.0
#define triger DDRB.0
#define OUT 1
#define INPUT 0
unsigned int CACAH=0;
unsigned jarak;
unsingned floating;

void force(void)
{
    PORTC.2=1;
    PORTC.3=0;
}

void turbo(void)
{
    PORTC.2=0;
    PORTC.3=1;
}

void transmitter (void)
{
    while(echo==0){ };
    while(echo==1)
        PULSA=0;
        triger=INPUT;
    delay_us(5);
        triger=OUT;
        PULSA=1;
        delay_ms(5000);
    }
    cacah++{ };
    float(jarak);
    jarak=((cacah)/344*2);
```

```

if (jarak <10){
force();
else{
turbo():
}
void main (void)
{
DDRC=0xFF;
while(1)
{
transmitter();
}
}

```

### 2.2.2 Sensor Jarak Ultrasonik Devantech SRF04

Sensor jarak merupakan sensor yang wajib ada pada robot terkini. Devantech SRF04 adalah salah satu sensor jarak yang paling banyak digunakan pada kontes robot di Indonesia selain ping Devantech. SRF04 ultrasonik range finder memberikan informasi jarak dari kisaran 3 cm – 3 m. Harga sensor ini tidak lebih dari Rp 360.000,00. Anda juga dapat membeli SRF05 yang harganya lebih murah dibandingkan SRF04 dengan kualitas yang tidak jauh berbeda.



Gambar 2.6 Devantech SRF04<sup>5</sup>

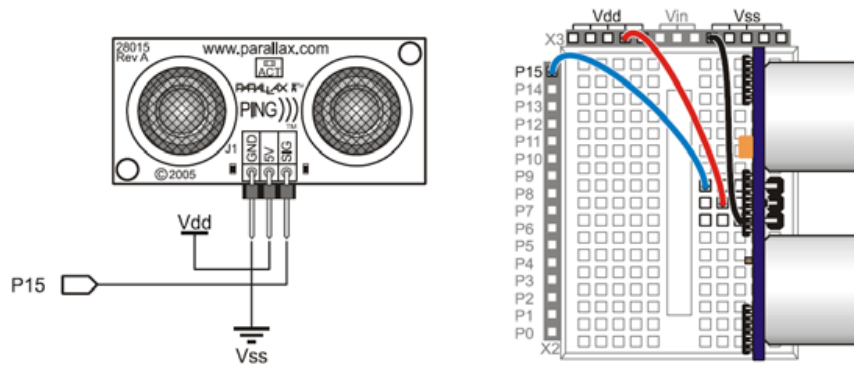
Kit ini sangat mudah untuk dirangkai dan membutuhkan sumber daya yang kecil sekali, yang sangat ideal untuk aplikasi mobil robot pencari jarak ini bekerja dengan cara memancarkan pulsa suara dengan kecepatan suara ( 0,9 ft/milidetik ).

---

<sup>5</sup> Ibid, hal.81



### 2.2.3 Instalasi Sensor Ultrasonic Ping



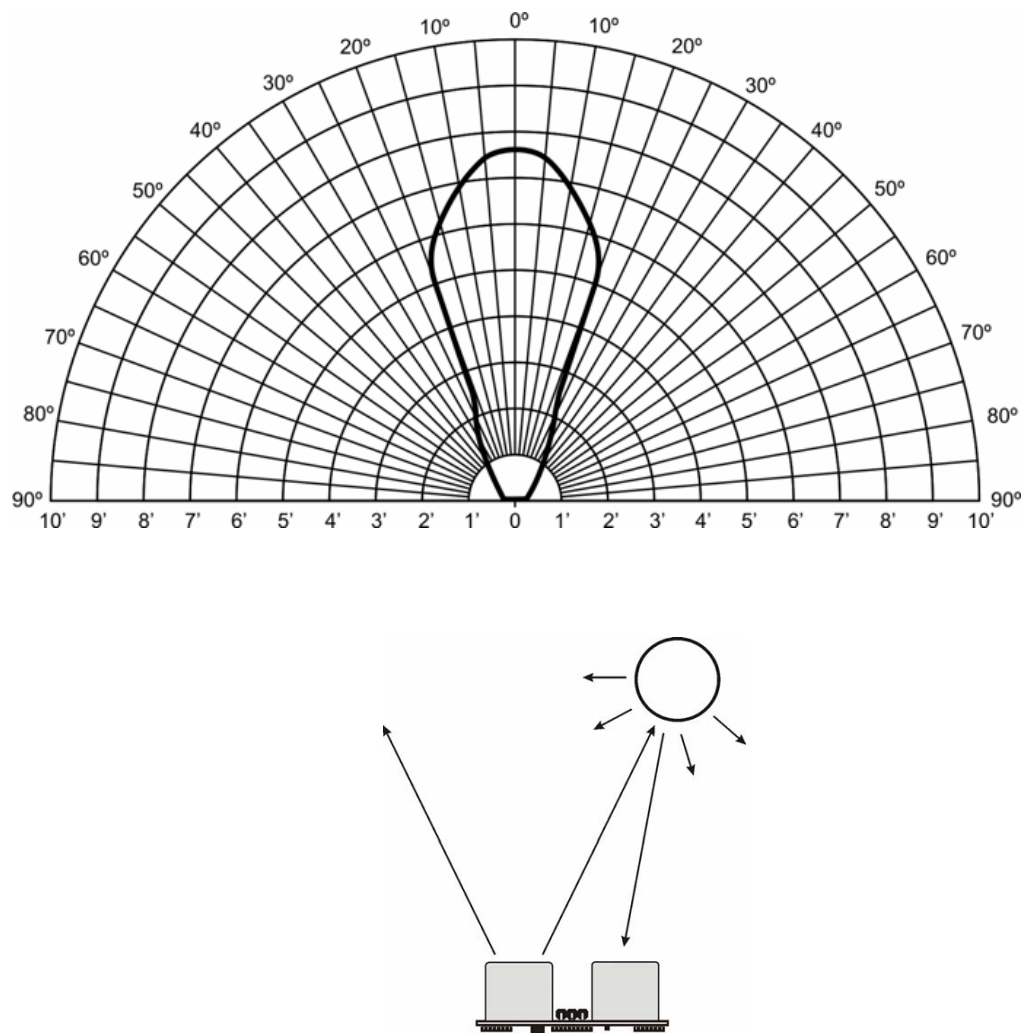
Gambar 2.7 Instalasi Sensor Ping<sup>6</sup>

Sensor ultrasonic ping akan bekerja jika mendapat suplay tegangan sebesar 5 V DC. dimana tegangan 5 V DC dihubungkan dengan konektor Vcc dan ground pada sensor. Untuk konektor SIG dapat dihubungkan dengan mikrokontroler. Konektor SIG adalah sebagai control sensor ini dalam pendeteksian objek sekaligus pembacaan jarak objek dengan sensor ini. progamer dapat mensetting sensor ini dengan jarak yang telah ditentukan sesuai dengan ring deteksi dari sensor ultrasonic ping ini sesuai dengan kebutuhan penggunaan dari sensor tersebut. Ketika sensor disetting jaraknya maka dengan jarak yang telah ditentukanlah sensor akan bekerja dalam pendeteksian objek. Kisaran jarak yang dapat di baca sensor ultrasonic ping ini adalah 3 cm sampai 3 m.

---

<sup>6</sup> ibid

Selain range jarak antara 3 cm sampai 3 m yang mampu dideteksi oleh sensor ultrasonik ping, sudut pancaran dari sensor jarak ultrasonic ping adalah dari  $0^\circ$  sampai dengan  $30^\circ$ .

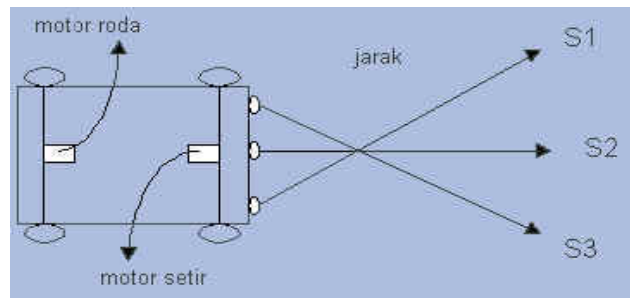


Gambar 2.8 Sudut Pancaran Sensor Ultrasonik Ping<sup>7</sup>

---

<sup>7</sup> ibid

#### 2.2.4 Prinsip Dasar dari Navigasi Robot



Gambar 2.9 Contoh mobil robot<sup>8</sup>

Motor roda yang terpasang pada bagian belakang dari robot mobil dipakai untuk mengatur kecepatan dari gerak maju / mundur robot mobil, sedangkan motor setir digunakan untuk mengontrol arah dan besar dari sudut belokan. Ada 3 buah sensor ultrasonik sebagai sensor jarak yang diletakkan di bagian depan dari robot mobil.

Peletakan sensor ultrasonik secara bersilangan dimaksudkan agar :

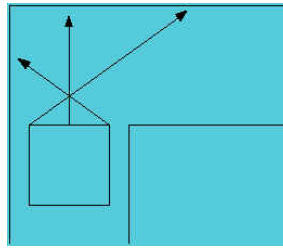
- Dapat mendeteksi besarnya halangan yang berada didepannya sehingga dapat menghindari halangan tersebut dengan baik.
- Masih mampu mendeteksi adanya belokan walaupun robot mobil sudah terlalu berdekatan dengan salah satu sisi dari jalur jalan
- Pendeteksian terhadap adanya belokan dari jarak yang masih jauh lebih baik karena mempunyai sudut pantulan yang lebih kecil jika dibandingkan dengan jika dipasang secara tidak bersilangan.

Perencanaan rules

---

<sup>8</sup> Sri Kusumadewi dan Hari Purnomo, *Aplikasi Logika Fuzzy untuk pendukung keputusan*, hal.55

Cara penentuan rules dapat dijelaskan sebagai berikut :



Gambar 2.10 Penentuan Rule Belok<sup>9</sup>

Pada kondisi ini diperoleh :

- 1 Sensor kanan mendeteksi jarak mobil dengan jalur kiri sangat dekat sehingga dapat dikatakan " Kanan is Very\_Near ".
- 2 Sensor kiri mendeteksi jarak mobil dengan jalur kanan sangat jauh (karena ada belokan) sehingga dapat dikatakan " Kiri is Very\_Far ".
- 3 Sensor tengah mendeteksi jarak mobil dengan jalur depan sangat dekat sehingga dapat dikatakan " Tengah is Very\_Near ".

Dari ketiga keadaan ini dapat dijadikan suatu rule yang dinyatakan sebagai :  
*" If Kanan is Very\_Near and Tengah is Very\_Near and Kiri is Very\_Far then Setir is Very\_Positif and Speed is Slow "*. Very\_Positif disini diartikan sebagai setir akan mempunyai prioritas belok kearah kanan dan Slow disini diartikan dengan kecepatan yang rendah

---

<sup>9</sup> Ibid, hal.77

### 2.3 MIKROKONTROLLER ATmega 8535

Mikrokontroller adalah suatu kombinasi mikroprosesor, piranti I/O (*Input/Output*) dan memori, yang terdiri atas ROM (*Read Only Memory*) dan RAM (*Random Access Memory*), dalam bentuk keping tunggal (*single chip*). Mikrokontroller ATmega8535 adalah mikrokontroller 8 *bit* buatan ATMEL dengan 8 KByte *System Programmable Flash* dengan teknologi memori tak sumirna (*nonvolatile*), kepadatan tinggi, dan kompatibel dengan pin out dan set instruksi standar industri MCS51 INTEL. Arsitektur yang digunakan dengan RISC (*Reduce Instruction set in single chip*). Mikrokontroller ATmega8535 memiliki karakteristik sebagai berikut :

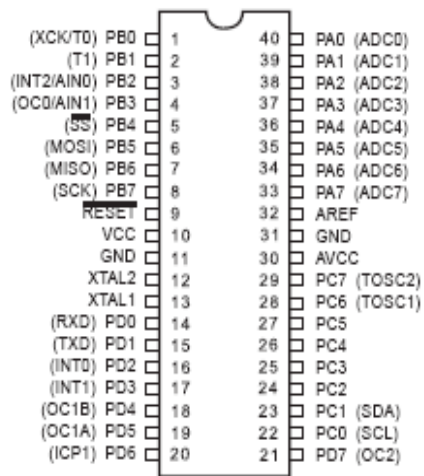
1. Kompatibel dengan produk keluarga MCS51.
2. Dapat digunakannya bahasa C sebagai bahasa pemrogramannya.
3. *Programmable Flash Memory* sebesar 8 K Byte.
4. Memiliki 512 Bytes EEPROM yang dapat diprogram.
5. Ketahanan (*endurance*) : 10.000 siklus tulis/hapus.
6. Jangkauan operasi : 4,5 – 5,5 Volt.
7. *Fully Static Operation* : 0 Hz – 16 MHz untuk ATmega8535.
8. Dua level *Program Memory Lock* yaitu *flash program* dan EEPROM *data security*
9. RAM Internal 128 X 8 *bit*,
10. Memiliki 32 jalur I/O yang dapat diprogram,
11. Satu pencacah 8 *bit* dengan *separate prescaler*,
12. Satu pencacah 16 *bit* dengan *separate prescaler*,
13. Sumber interupsi (*interrupt source*) eksternal dan internal,

14. Kanal pengirim-penerima tak serempak universal (UART-*UniversalAsynchronous Receiver-Transmitter*) yang dapat diprogram,

15. *Low-power Idle* dan *Power-down Model*

### 2.3.1 Susunan Kaki MIKROKONTROLER ATmega8535

Bentuk kemasan dan susunan kaki-kaki mikrokontroler dari ATmega8535 diperlihatkan seperti pada Gambar 2.11



Gambar 2.11 Susunan kaki pada ATmega8535.<sup>10</sup>

Penjelasan dari masing-masing kaki adalah sebagai berikut:

1. VCC (kaki 40) dihubungkan ke Vcc
2. GND (kaki 20) dihubungkan ke *ground*.
3. PortA (PA7..PA0) (kaki 32-39) merupakan port 8 bit dua arah (*bidirectional*) I/O. Port ini berfungsi sebagai port data/alamat I/O ketika menggunakan SRAM eksternal.
4. Port B (PB7..PB0) (kaki 1-8) merupakan port 8 bit dua arah (*bidirectional*) I/O, untuk berbagai keperluan (*multi purpose*).

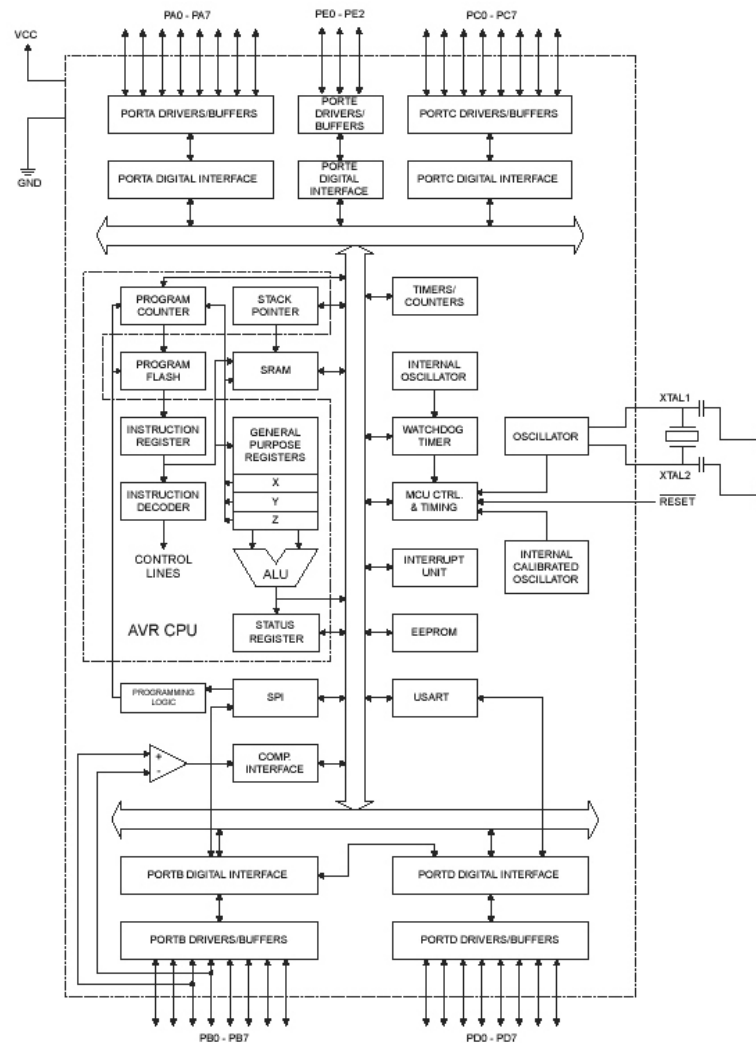
<sup>10</sup> Agus Bejo, *Kemudahan C dan mikrokontroler AVR*, hal.4

5. Port C (PC7..PC0) (kaki 21-28) adalah port 8 bit dua arah I/O, dengan *internal pull-up* resistor. Port C ini juga berfungsi sebagai port alamat ketika menggunakan SRAM eksternal.
6. Port D (PD7..PD0) (kaki 10-17) adalah port 8 bit dua arah I/O dengan resistor *pull-up* internal. Port D juga dapat berfungsi sebagai terminal khusus.
7. *Reset* (kaki 9) ketika kondisi rendah rendah yang lebih lama dari 50 nS mikrokontroler akan *reset* walaupun detak tidak berjalan.
8. XTAL1 (kaki 19) masukan bagi penguat osilator terbalik dan masukan bagi rangkaian operasi detak internal.
9. XTAL2 (kaki 18) keluaran dari penguat osilator terbalik.
10. ICP (kaki 31) adalah masukan bagi masukan fungsi *Capture Timer/counter1*.
11. OC1B (kaki 29) adalah kaki keluaran bagi fungsi *Output CompareB* keluaran *Timer/Counter1*.
12. ALE (*Address Latch Enable*) (kaki 30) digunakan ketika menggunakan SRAM eksternal. Kaki ini digunakan untuk mengunci 8 bit alamat bawah pada saat siklus akses pertama, dan berfungsi sebagai port data pada siklus akses kedua.

### 2.3.2 Blok Diagram dan Arsitektur ATmega8535

ATmega8535 mempunyai 32 *general purpose register* (R0..R31) yang terhubung langsung dengan *Arithmetic Logic Unit* (ALU), sehingga *register* dapat diakses dan dieksekusi hanya dalam waktu satu siklus clock. ALU merupakan tempat dilakukannya operasi fungsi aritmetik, logika dan operasi *bit*. R30 disebut juga sebagai *Z-Register*, yang digunakan sebagai *register* penunjuk pada pengalamatan tak langsung. Didalam ALU terjadi operasi aritmetik dan logika

antar *register*, antara *register* dan suatu konstanta, maupun operasi untuk *register* tunggal (*single register*). Berikut arsitekturnya yang ditunjukkan blok diagram pada Gambar 2.12



Gambar 2.12 Diagram blok Mikrokontroler ATmega8535.<sup>11</sup>

### 2.3.3 Organisasi Memori

AVR menggunakan arsitektur Harvard, sehingga memisahkan memori serta bus data dengan program. Program ditempatkan *Flash Memory*, sedangkan

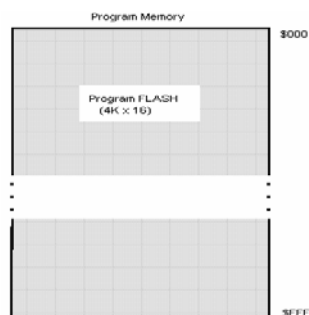
<sup>11</sup> [www.ATMEL.com>Data](http://www.ATMEL.com>Data) sheet ATmega 8535, Minggu, 8 Juni 2008, jam.09.10



memori data terdiri dari 32 buah register serbaguna, 64 register serbaguna, 512 bytes internal SRAM dan 64 Kbytes SRAM eksternal yang dapat ditambahkan. Berdasarkan fungsinya terdapat 4 macam memori pada ATmega8535.

#### 2.3.4 Memori Program

ATmega8535 mempunyai kapasitas memori program sebesar 8 Kbytes. Karena semua format instruksi berupa kata (word), Format word yang biasa digunakan adalah 16 atau 32 bit. Pada ATmega8535 ini format memori program yang digunakan adalah 16 bit, sehingga format memori program yang digunakan adalah 4Kx16bit. Memori Flash ini dirancang untuk dapat di hapus dan tulis sebanyak seribu kali. *Program Counter* (PC)-nya sepanjang 12 bit, sehingga mampu mengakses hingga 4096 alamat program memori. Memori program pada ATmega8535 diperlihatkan pada Gambar 2.13

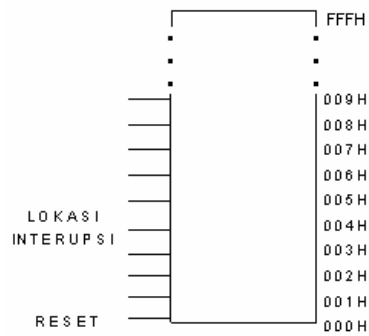


Gambar 2.13 Memori program.<sup>12</sup>

Gambar 2.14 memperlihatkan bagian bawah dari memori program. Setelah *reset* CPU memulai eksekusi dari lokasi 0000h. Setiap interupsi mempunyai lokasi tetap dalam memori program. Interupsi menyebabkan CPU melompat ke lokasi tersebut dimana pada lokasi tersebut terdapat subrutin yang harus dilaksanakan.

---

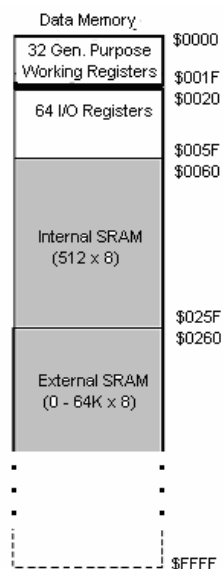
<sup>12</sup> [www.ATMEL.com.Data](http://www.ATMEL.com/Data) sheet ATmega 8535, Minggu, 8 Juni 2008, jam.09.10



Gambar 2.14 Bagian bawah memori program<sup>13</sup>

### 2.3.5 Memori Data

Memori data pada ATmega8535 diperlihatkan pada Gambar 2.15. Memori data pada ATmega 8535 tersebut terdiri atas memori data internal dan eksternal kapasitas dari mSRam internal adalah sebesar 512 Kbytes. ini menempati ruang alamat setelah 32 lokasi register serbaguna. alamat dan setelah 64 register I/O. Jika SRAM eksternal digunakan, ini akan mengikuti besar SRAM eksternal sampai dengan maksimum 64K tergantung ukuran SRAM eksternal



Gambar 2.15 Memori Data<sup>14</sup>

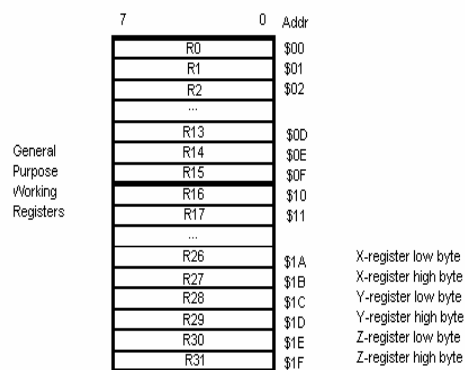
<sup>13</sup> [www.ATMEL.com.Data](http://www.ATMEL.com/Data) sheet ATmega 8535, Minggu, 8 Juni 2008, jam.09.10

<sup>14</sup> [www.ATMEL.com.Data](http://www.ATMEL.com/Data) sheet ATmega 8535, Minggu, 8 Juni 2008, jam.09.10

Ketika alamat yang digunakan untuk mengakses melebihi alamat dari ruang memori internal SRAM, maka SRAM eksternal akan terakses dengan instruksi yang sama dengan instruksi pada SRAM internal. Jalur pengontrolan SRAM untuk baca dan tulis masing-masing menggunakan kaki *RD* dan *WR*. Ketika masih mengakses SRAM internal, fungsi *RD* dan *WR* tidak akan aktif dan kaki tersebut berfungsi seperti biasa. Operasi SRAM eksternal dimungkinkan oleh setting bit SRE di register MCUCR. Mode pengalamatan untuk mengakses memori data meliputi pengalamatan langsung (*Direct Addressing*), pengalamatan tak langsung (*Indirect Addressing*), *Indirect* dengan *Pre-Decrement*, *Indirect* dengan *Post-Decrement*, dan *Indirect* dengan *Displacement*.

### 2.3.6 Register Serbaguna

Pada 32 *bytes general purpose working register* atau register umum serbaguna mendukung adanya konsep register akses cepat. Ini berarti bahwa waktu akses dari register adalah satu detak dan berarti pula satu operasi ALU (Arithmetics Logic Unit) dilaksanakan. Agar lebih jelas, maka 32 register itu ditunjukkan pada Gambar 2.16

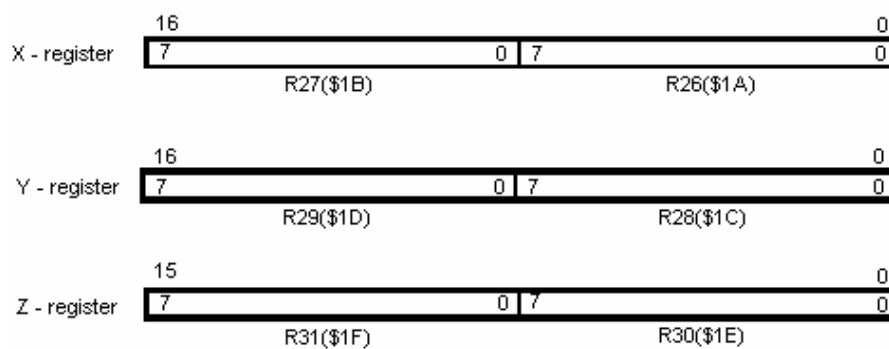


Gambar 2.16 Pemetaan Memori 32 Register Serbaguna<sup>15</sup>

<sup>15</sup> [www.ATMEL.com.Data](http://www.ATMEL.com>Data) sheet ATMEga 8535, Minggu, 8 Juni 2008, jam.09.10

Semua instruksi operasi register mempunyai akses langsung dan siklus tunggal terhadap semua register. Perkecualiannya adalah pada perintah SBCI, SUBI, CPI, ANDI dan ORI. Instruksi ini digunakan pada R16 sampai dengan R31. Enam dari 32 register dapat digunakan sebagai tiga penunjuk alamat tak langsung 16 bit untuk pengalamatan ruang data. Tiga register fungsi tambahan ini disebut register 16 bit X,Y, dan Z.

Register di atas juga bisa dianggap sebagai alamat memori data. Register X,Y, dan Z dapat diset sebagai index pada berbagai register. Register X, Y dan Z tersebut ditunjukkan pada Gambar 2.17

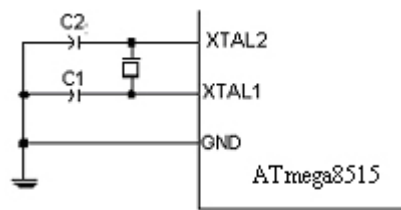


Gambar 2.17 Register X, Y dan Z.<sup>16</sup>

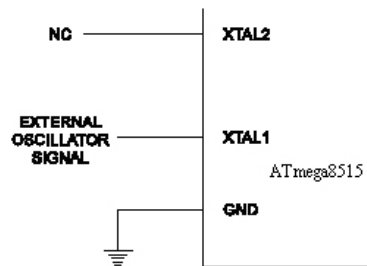
### 2.3.7 Pewaktuan CPU

Mikrokontroler ATmega8535 memiliki osilator internal (*on chip osilator*) yang dapat digunakan sebagai sumber detak bagi CPU. Untuk menggunakan osilator internal diperlukan sebuah kristal atau resonator keramik antara kaki Xtal1 dan kaki Xtal2 dan dua buah kapasitor yang ditambahkan seperti terlihat pada Gambar 2.18.

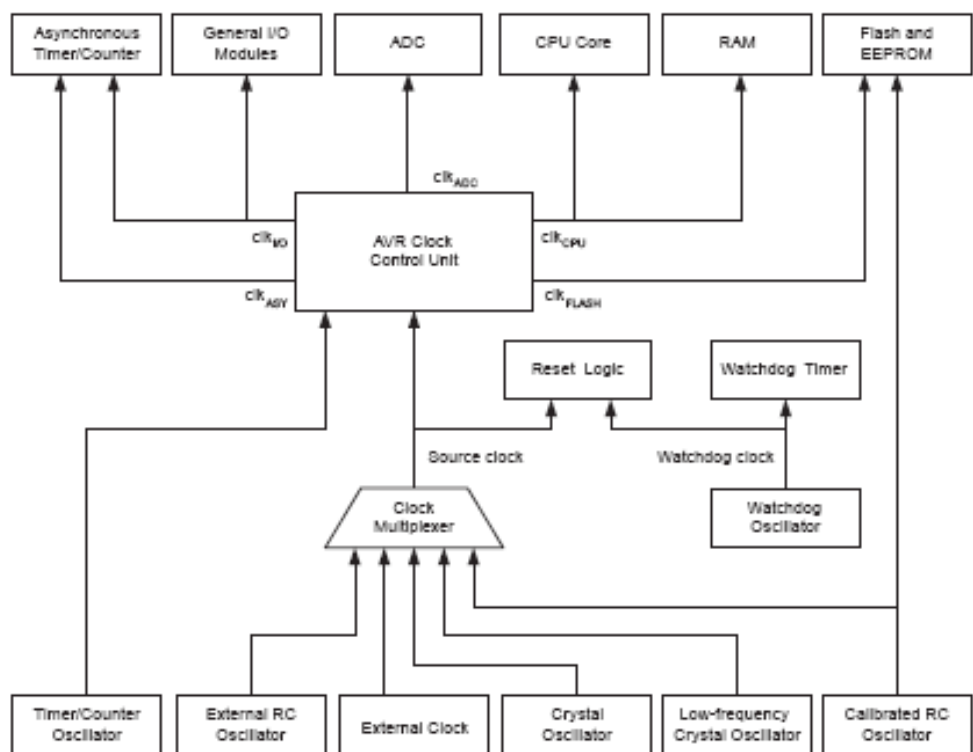
<sup>16</sup> [www.ATMEL.com.Data](http://www.ATMEL.com/Data) sheet ATmega 8535, Minggu, 8 Juni 2008, jam.09.10



Gambar 2.18 Menggunakan osilator internal<sup>17</sup>



Gambar 2.19 Menggunakan sumber detak eksternal<sup>18</sup>



Gambar 2.20 Distribusi pewaktu ( clock )<sup>19</sup>

<sup>17</sup> [www.ATMEL.com.Data](http://www.ATMEL.com.Data) sheet ATmega 8535, Minggu, 8 Juni 2008, jam.09.10

<sup>18</sup> Agus Bejo, *Kemudahan C dan mikrokontroler AVR*, hal.11

<sup>19</sup> Ibid, hal.12

### 2.3.8 PORT A/B/C/D/E, DDR A/B/C/D/E, DAN PIN A/B/C/D/E

PORTA/B/C/D/E dan DDRA/B/C/D/E merupakan *register-register* yang digunakan untuk mengatur PORTA/B/C/D/E, sedangkan PIN PORTA/B/C/D/E digunakan untuk mengakses pin pada *port* A,B,C,D,E secara individu. Hubungan antara PORT PORTA/B/C/D/E dan DDR PORTA/B/C/D/E diperlihatkan pada tabel 2.1

Tabel 2.1 Kombinasi *bit* DDRA/B/C/D/En dan PORTA/B/C/D/En .

DDRBn/Dn	PORTBn/Dn	I/O	Keterangan
0	0	<i>Input</i>	<i>Tri-state (High-Z)</i>
0	1	<i>Input</i>	PORTA/B/C/Dn akan menghasilkan arus jika eksternal <i>pull-low</i>
1	0	<i>Output</i>	<i>Push-pull zero output</i>
1	1	<i>Output</i>	<i>Push-pull one output</i>

www. ATMEL. *Data Sheet Book AT Mega8535*, h.8, Kamis, 15 Mei 2008

### 2.3.9 Pewaktu atau Pencacah

Pencacah pada ATmega8535 diatur oleh register TCCR1B (*Timer/Counter1 Control Register B*). Register TCCR1B dijelaskan pada Gambar 2.21

Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.21 Register TCCR1B.<sup>20</sup>

<sup>20</sup> Ibid, hal.16

Penjelasan dari masing-masing bit adalah sebagai berikut:

- Bit 7 ICNC1 (*Input Capture1 Noise Canceler* (4 CKs)

Bit ini berfungsi meng-*enable/disable* fungsi *noise canceler*

- Bit 6 ICES1 (*Input Capture1 Edge Select*)

Bit ini berfungsi memilih jenis tepian untuk memicu ICP

- Bit 4:3 WGM13:2 (*Waveform Generation Mode*)

Bit ini berfungsi mengaktifkan fungsi *Waveform Generation*.

- Bit 2:0 CS12, CS11, CS10: *Clock Select1*, Bits 2, 1 dan 0

Kombinasi dari bit-bit ini menentukan sumber *prescale* dari *Timer/Counter* sebagaimana dijelaskan melalui Tabel 2.2

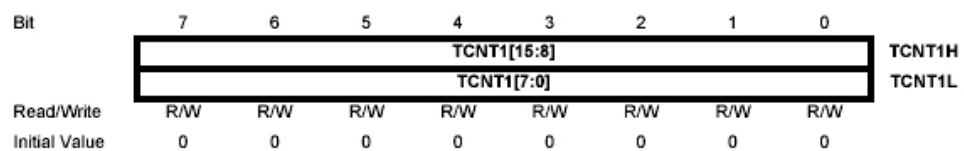
Tabel 2.2 Clock 1 *Prescale Select*.

CS12	CS11	CS10	Deskripsi
0	0	0	<i>Stop, Timer/Counter1</i> dihentikan
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	Kaki Eksternal T1, tepian turun
1	1	1	Kaki Eksternal T1, tepian naik

www. ATMEL. *Data Sheet Book AT Mega8535*, h.8, Kamis, 15 Mei 2008

Kondisi *stop* memberikan fungsi *enable/disable*. CK dan CK terbagi menentukan frekuensi detak yang digunakan *Timer/Counter1*. Eksternal pin T1 *falling/rising edge* memilih jenis tepian yang digunakan untuk memicu *counter* yang diberikan pada kaki T1 (kaki nomor 2).

Register TCNT1 merupakan register yang berisi data 16-bit hasil perhitungan pencacah. Register ini memiliki fungsi akses langsung, baik untuk operasi menulis atau membaca data. Register TCNT1 terbagi atas 2 register 8-bit yaitu TCNT1H dan TCNT1L seperti pada gambar 2.22



Gambar 2.22 Register TCNT1<sup>21</sup>

## 2.4 Program Pendukung

Pada pembuatan program mikrokontroler, memerlukan suatu sistem program untuk menempatkan dan mengirim program dari PC ke mikrokontroler. Sistem program pendukung yang digunakan pada modul ini adalah *code vision AVR versi avaluation*.

### 2.4.1 Code Vision AVR Full Version 1.24.2 Series

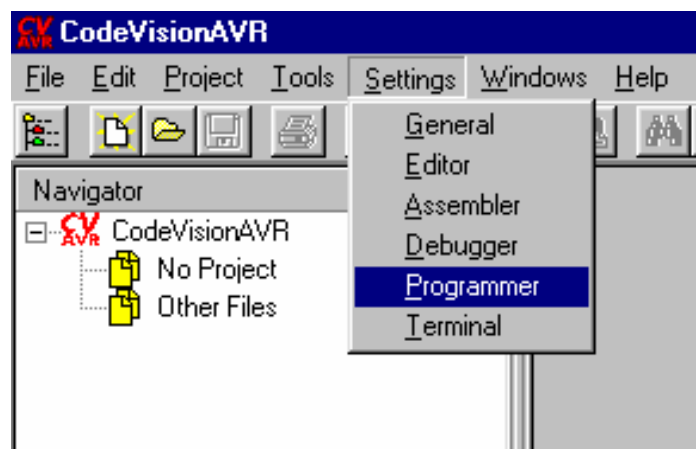
Code vision AVR merupakan salah satu program bahas C yang berbasis Windows, keuntungan menggunakan code vision AVR lebih besar dibandingkan menggunakan program yang lain yang under DOS. Code vision AVR dalam pemrogramannya menggunakan bahasa C maupun bahasa C++. Namun dalam pembuatan Tugas Akhir penulis menggunakan code vision AVR untuk bahasa C.karena bahasa C sangat compatibel dengan mikrokontroller AVR terutama mikrokontroller ATMega 8535

<sup>21</sup> Ibid, hal.18



Code vision AVR menyediakan area kerja dan toolbar yang mudah untuk melakukan berbagai operasi. Code vision AVR memiliki beberapa menu aplikasi windows yaitu meliputi File, Project, Edit, Debug, View, Tool, Windows, Help, check sytak eror compile dan make.

Ada perbedaan pada menu Code Vision AVR yaitu, ketika belum ada satu pun project atau file yang dibuka, Code Vision AVR hanya memiliki enam menu yaitu File, Project, View, Tools, Debug dan Help.

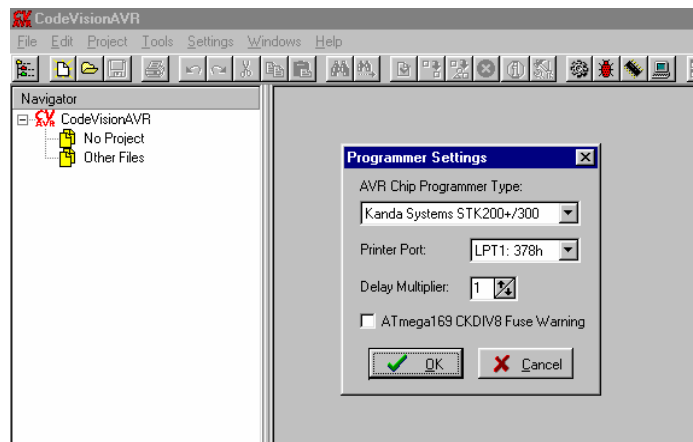


Gambar 2.23 Tampilan Code Vision AVR<sup>22</sup>

Namun jika sebuah project atau file telah dibuka, atau sedang dikerjakan, maka akan terdapat dua menu tambahan yaitu Edit dan Windows, seperti gambar 2.24 di bawah ini.

---

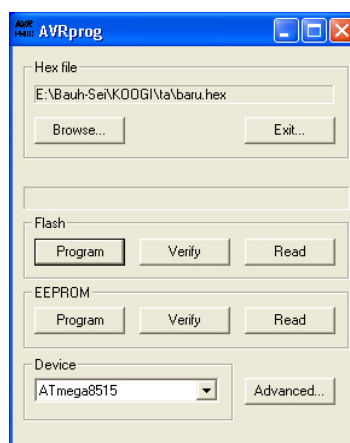
<sup>22</sup> M Haryanto, Pemrograman Mikrokontroler Atmega 8535 dengan bahasa C. hal.34



Gambar 2.24 Tampilan Code vision AVR<sup>23</sup>

#### 2.4.2 ISP (In-System Programming)

ISP (In-System Programming) merupakan program khusus yang digunakan untuk menghubungkan antara PC dengan mikrokontroller ATmega 8535. Program yang digunakan adalah AVRprog berfungsi untuk mentransfer program yang telah dibuat dari PC melalui program Code Vision AVR ke flash memori pada mikrokontroller ATmega 8535. Tampilan dari ISP AVRprog dapat dilihat pada gambar 2.25.



Gambar 2.25 Tampilan AVRprog<sup>24</sup>

<sup>23</sup> Ibid, hal.36

Pada tampilan gambar AVRprog di atas ada beberapa instruksi yang dibutuhkan untuk melakukan proses pemasukan program ke flash memori mikrokontroller yaitu:

1. Browse berfungsi untuk mencari alamat program yang telah ditulis
2. Exit berfungsi untuk keluar dari program AVRprog
3. Program berfungsi untuk memasukan program yang telah ditulis ke flash memori
4. Verify berfungsi untuk memilih program yang akan masuk ke flash memori
5. Read berfungsi untuk membaca program yang telah ditulis
6. Advanced berfungsi untuk memilih tipe mikrokontroller yang digunakan

## 2.5 Catu Daya

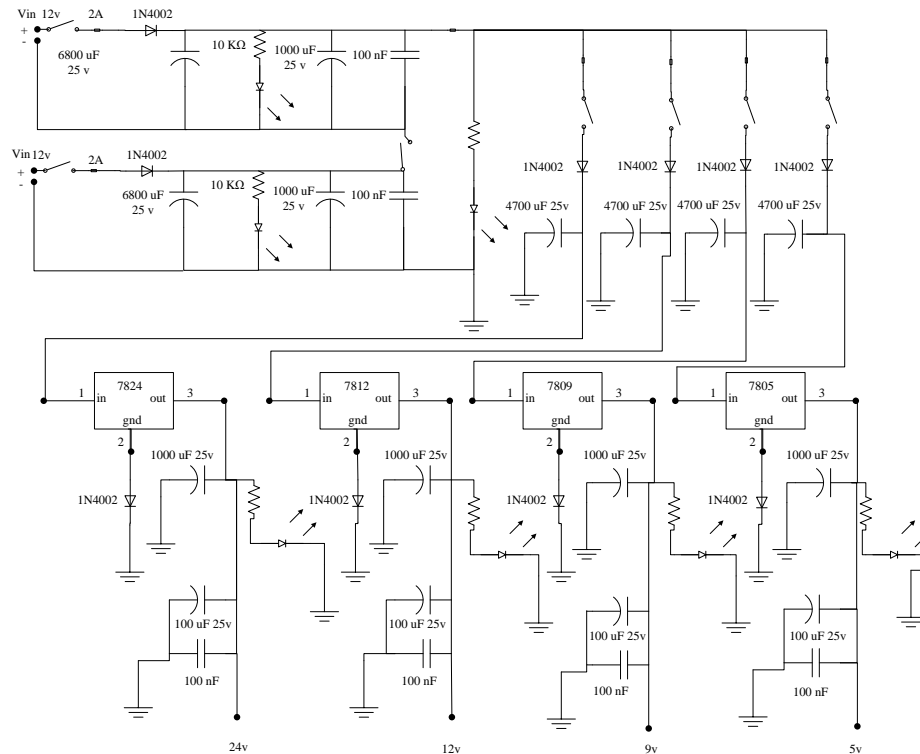
Rangkaian catu daya merupakan rangkaian yang mendukung bekerjanya sistem ini. Rangkaian catu daya ini tidak akan dibahas secara mendalam karena tidak termasuk pokok bahasan.

Perangkat elektronika mestinya dicatu oleh suplai arus searah DC (*direct current*) yang stabil agar dapat bekerja dengan baik. Baterai atau accu adalah sumber catu daya DC yang paling baik. Pada tugas akhir ini sumber daya yang digunakan adalah baterai, keluaran (out put )dari baterai ini adalah sudah berupa tegangan DC (*direct current*).tetapi tegangan yang dikeluarkan oleh baterai ini belum konstan, oleh karena itu diperlukan penstabil tegangan agar tegangan

---

<sup>24</sup> Ibid, hal.37

keluaran dari baterai ini dapat stabil. Secara blok catu daya dapat digambarkan sebagai berikut :



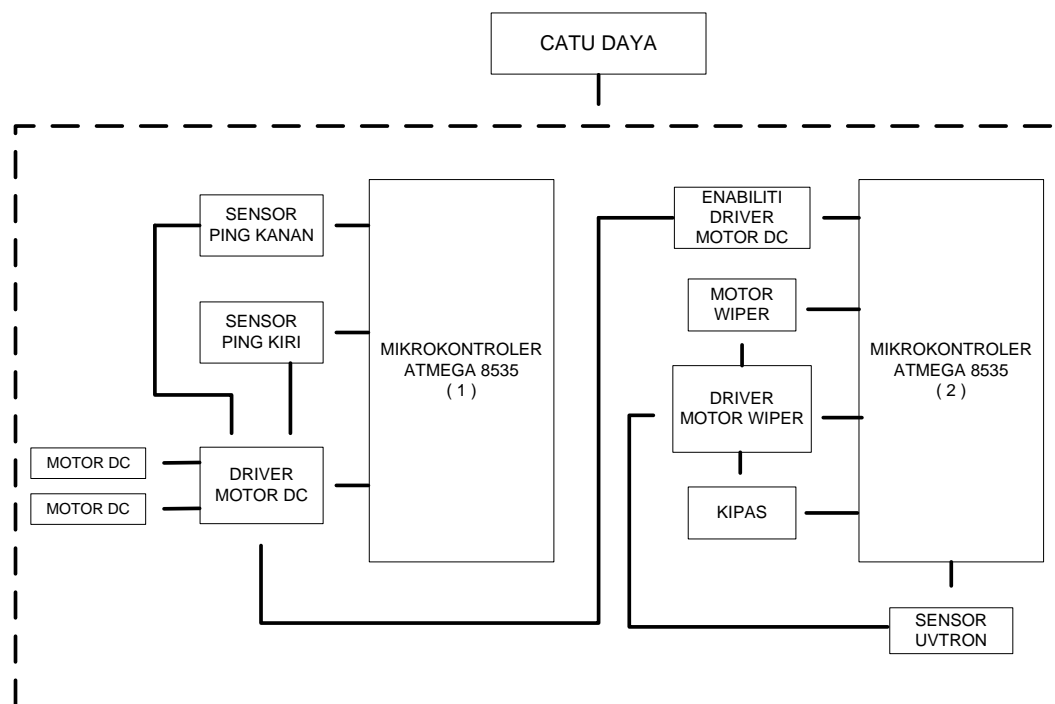
Gambar 2.26 Catu Daya

Pada gambar diatas terlihat beberapa output dari catu daya yang digunakan. semua tegangan out put tersebut digunakan untuk mensuplai tegangan semua rangkaian dari robot pemadam api. 24 volt digunakan untuk mensuplai diver motor DC sehingga mampu menggerakkan motor Dc. 12 volt digunakan untuk mensuplay motor washer agar mampu memadamkan nyala api. 9 volt digunakan mentrigger driver sensor UVtron agar dapat bekerja mendeteksi nyala api serta volt digunakan untuk mensuplai mikrokontroler agar mikrokontroler dapat bekerja. 5 volt digunakan untuk mensuplai sensor ping agar ping dapat mendeteksi ahalan yang dapat menghambat laju robot.

## BAB III

### PRINSIP KERJA RANGKAIAN

Bab ini membahas prinsip kerja rangkaian yang disusun untuk merealisasikan sistem alat dalam hal ini sensor ultrasonik sebagai sensor pendeteksi penghalang agar robot pemadam api dapat menghindari tabrakan dengan dinding. Adapun sistem alat yang dibuat dan dirancang sesuai blok diagram dibawah ini. pembahasan dititik beratkan pada perancangan alat yang dibuat berdasarkan pemikiran penulis mengacu pada sumber acuan yang berhubungan dengan alat, khususnya bagian sensor ultrasonik pada robot pemadam api berbasis mikrokontroller ATmega 8535.

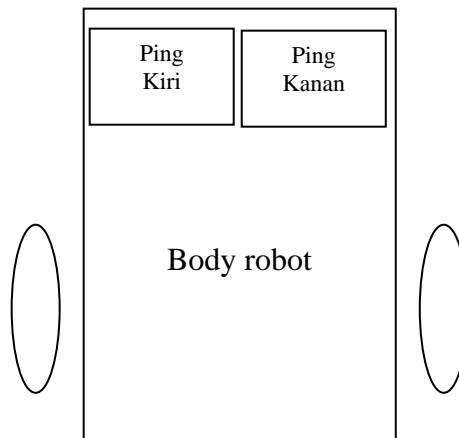


Gambar 3.1 Blok diagram robot pemadam api

### 3.1 Prinsip Kerja Tiap Blok

prinsip kerja tiap blok akan menggambarkan cara kerja rangkaian tiap bagian diantaranya akan menerangkan cara kerja dari sensor ultrasonik sebagai alat navigasi pada robot,, catu daya dan program code vision AVR versi evaluation yang digunakan dalam pembuatan software.

#### 3.1.1 Cara Kerja Sensor Ultrasonik Sebagai Navigasi Robot



Gambar 3.2 Letak sensor ultrasonik

Fungsi dari sensor ultrasonik tersebut adalah sebagai alat navigasi pada robot. Dimana dengan sensor ultrasonik inilah robot akan berjalan dan mampu menghindari halangan yang menghalangi laju robot sehingga robot tidak menabrak. Prinsipnya cukup mudah dimana semua transceiver sensor ultrasonik samping kiri dan samping kanan memancarkan gelombang ultrasonik untuk mendeteksi adanya halangan yang dapat menghambat laju robot, gelombang ultrasonik tersebut kemudian ditangkap oleh bagian receiver pada sensor ultrasonik, setelah receiver pada sensor ultrasonik menangkap pancaran

gelombang yang mendeteksi adanya halangan selanjutnya semua sensor ultrasonik baik samping kiri dan samping kanan memberikan data kepada mikrokontroller untuk memerintahkan logika apa dan apa yang harus dilakukan oleh robot apakah belok kiri, jalan lurus atau belok kanan.

Berikut adalah logika listing software ultrasonik

```
/*Master of Software development
   Delta_^production_general software*/
#include <mega8535.h>
#include <delay.h>
#define PULSA PORTB.0
#define echo PINB.0
#define triger DDRB.0
#define PULSA_1 PORTB.2
#define echo_1 PINB.2
#define triger_1 DDRB.2
#define OUT 1
#define INPUT 0
unsigned int CACAH=0;
unsigned jarak;
unsigned floating;

void force(void)
{
  PORTC.2=1;
  PORTC.3=0;
}

void turbo(void)
{
  PORTC.2=0;
  PORTC.3=1;
}

void force_1(void)
{
  PORTC.4=1;
  PORTC.5=0;
}

void turbo_1(void)
{
  PORTC.4=0;
  PORTC.5=1;
}
```

```

void transmitter (void)
{

while(echo==0){ };
while(echo==1)
    PULSA=0;
    triger=INPUT;
delay_us(5);
    triger=OUT;
    PULSA=1;
    delay_ms(5000);
}
cacah++;{ };
float(jarak);
jarak=((cacah)/344*2);
if (jarak <10){
force();
else{
turbo():
}
}
void transmitter_2 (void)
{

while(echo_1==0){ };
while(echo_1==1)
    PULSA_1=0;
    triger_1=INPUT;
delay_us(5);
    triger_1=OUT;
    PULSA_1=1;
    delay_ms(5000);
}
cacah++;{ };
float(jarak);
jarak=((cacah)/344*2);
if (jarak <10){
force_1();
else{
turbo_1():
}
}
void main (void)
{
DDRC=0xFF;
while(1)
{
transmitter();
transmitter_1();
}
}

```

List diatas mengartikan jika sensor ultrasonik mendeteksi adanya halangan kemudian memberikan data pada mikrokontroller untuk melihat kondisi pada

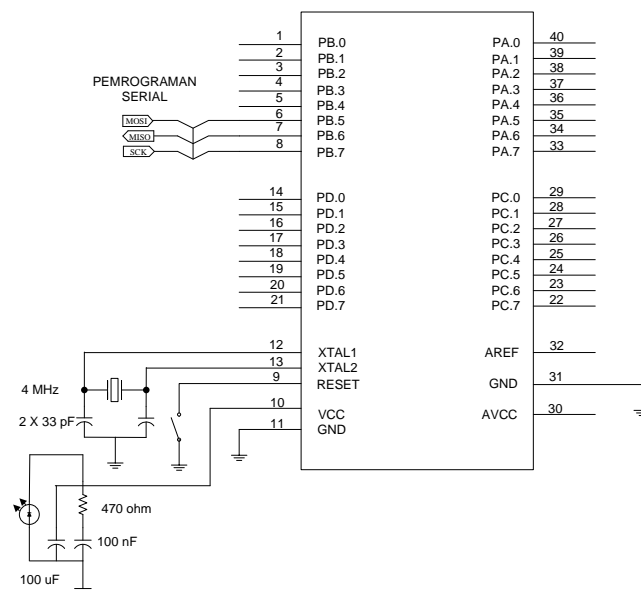


sensor kiri dan kanan, jika kiri sempit dan kanan luas maka robot akan berbelok ke kanan dan jika kanan sempit dan kiri luas robot akan berbelok ke kiri.

Logika sensor ultrasonik sebagai alat navigasi robot

- Jika tidak ada halangan di depan maka robot maju terus
- Jika di depan terdapat halangan dan kiri lebih sempit dari kanan maka robot akan mundur dulu untuk kemudian berbelok ke kanan
- Jika di depan terdapat halangan dan kanan lebih sempit dari kiri maka robot mundur dulu untuk kemudian berbelok ke kiri
- Jika di depan ada halangan, kiri ada halangan, kanan tidak ada halangan maka mundur dulu belok kanan
- Jika di depan ada halangan, kanan ada halangan, kiri tidak ada halangan maka mundur dulu belok kiri

### 3.1.2 Cara Kerja Mikrokontroller ATmega 8535



Gambar 3.3 Rangkaian Sistem minimum mikrokontroller ATmega 8535

Mikrokontroller bertugas melakukan proses deteksi data masukan, mengolah data dan mengatur keluaran sesuai dengan fungsi alat yang dikehendaki. Pada perancangan ini mikrokontroller yang digunakan adalah ATmega 8535 yang merupakan keluarga mikrokontroller dari ATMEL.

Pada bagian ini mikrokontroller dihubungkan dengan beberapa perangkat eksternal baik itu sebagai masukan maupun keluaran, dimana mikrokontroller akan mengendalikan semua aktivitas robot dalam usaha untuk memadamkan api berdasarkan data yang diterima dari masukan.

Berikut ini merupakan hubungan kaki – kaki mikrokontroller dengan perangkat luar yaitu sebagai berikut :

#### Microkontroler 2

Nama kaki	Keterangan
Pc.4	dihubungkan ke driver motor Washer untuk menggerakkan motor Washer dan kipas
Pc.0 dan pc.7	dihubungkan ke enablitasi pada driver motor DC
Pb.0	dihubungkan ke driver ultrasonik agar dapat mendeteksi nyala api

#### Microkontroler 3

Nama kaki	Keterangan
Pa.0	dihubungkan ke ping samping kiri untuk mendeteksi adanya halangan di samping kiri robot
Pa.2	dihubungkan ke ping samping kanan untuk mendeteksi adanya halangan di samping kanan robot
Pc.2 – Pc.5	dihubungkan ke driver motor DC L298D untuk menggerakkan motor DC supaya robot dapat berjalan

Mikrokontroler mendapat trigger sebesar 9 volt DC dari power supply sehingga membuat sistem minimum mikrokontroler ini menjadi aktif. Sistem kerja dari perangkat mikrokontroler ATmega 8535 ini adalah sebagai berikut ini. dalam sistem ini terdiri dari dua buah mikrokontroler dimana masing - masing mikrokontroler difungsikan untuk memberi pulsa pada sensor-sensor yang terhubung pada I/O. mikrokontroler 2 berfungsi untuk mengatur enabilti motor DC, motor washer dan sensor UVtron. Dimana jika tidak terdapat nyala api maka mikrokontroler 2 akan memberi logika 1 pada enabilti yang terpasang pada Pc.0 dan Pc.7 membuat enabilti pada driver motor DC menjadi aktif. Mikrokontroler 2 berfungsi untuk mengatur Motor DC, dan Sensor Ultrasonic Ping. Sensor ultrasonic berfungsi sebagai alat navigasi pada robot dimana terdiri dari Ping samping kiri dan samping kanan. Jika terdapat halangan pada sensor maka sensor akan memberi logika 1 untuk diberikan pada mikrokontroler kemudian mikrokontroler mengolahnnya untuk dikeluarkan ke Motor DC. Jika terdapat nyala api maka logika dari sensor UVtron akan high mengakibatkan enabilti pada motor low dan motor akan berhenti. Dengan delay 5 detik maka mikrokontroler 2 akan mentrigger driver motor washer dan memerintahkan motor washer untuk memadamkan nyala api tersebut.

### **3.1.3 Program Code Vision AVR Full Version 1.24.2 Series**

Berbeda dengan Code Vision AVR Full Version Evaluation dimana versi Full ini dapat mencompile file .c kedalam file .HEX sebesar 8 kbyte sementara jika kita menggunakan versi evaluation hanya mampu menghasilkan file.HEX sebesar 2 kbyte, hal ini membuat kita dapat menciptakan suatu program dengan

kemampuan yang cukup tinggi karena didukung dengan memori penampungan file yang cukup besar

Pada Code Vision AVR Program yang ditulis dalam bahasa C disebut program sumber (*source program*) yang biasanya disimpan dalam file berekstensi .C. Setelah melalui proses compile dan built make akan dihasilkan satu atau lebih output sesuai dengan konfigurasi pada saat proses compile dijalankan. File-file output hasil proses compile antara lain, assembly , .inc prj dan hexa yang kemudian disebut sebagai Object File.

Object File merupakan file yang berisi kode object yang secara langsung akan dijalankan oleh mikrokontroler. Object kode yang akan dijalankan mikrokontroler berbentuk kode biner yang akan diisikan ke EEPROM. Sedangkan object kode yang akan ditransfer ke peralatan lain biasanya berbentuk kode hexa. Kode hexa ada beberapa standart diantaranya adalah intel HEX dan motorolla S1.HEX.

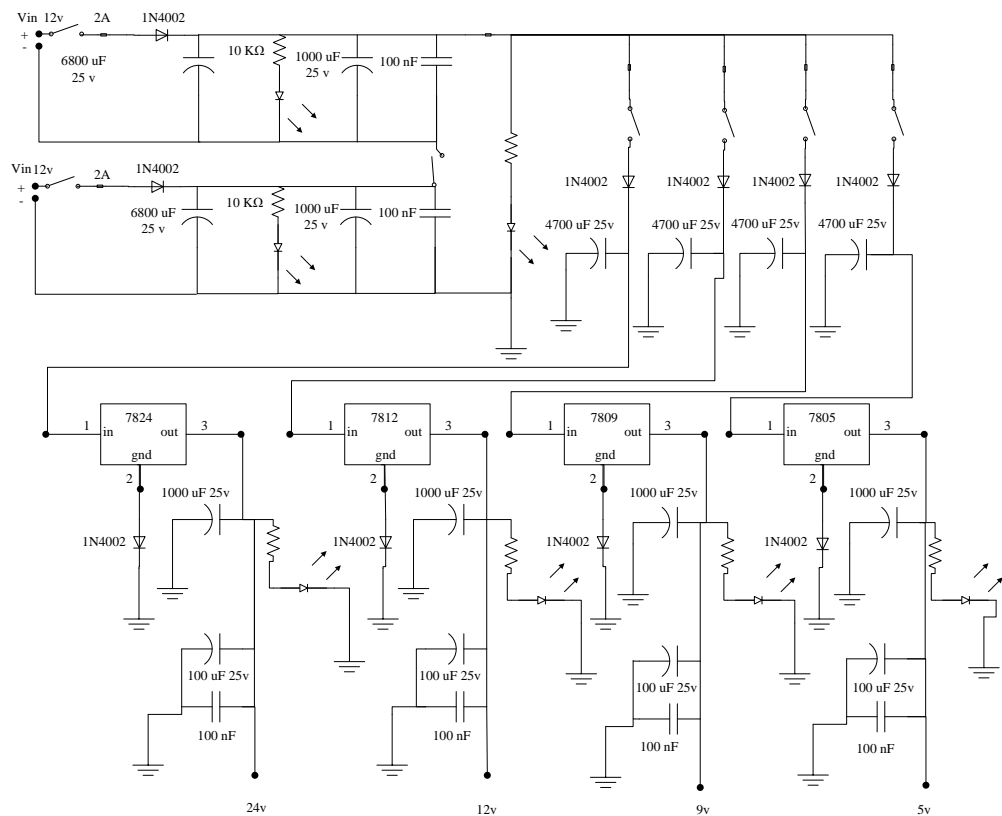
#### **3.1.4 Catu Daya**

Catu daya pada rangkaian robot pemadam api ini adalah menggunakan sumber tegangan dc.jadi tidak terlalu susah karena tinggal menstabilkan tegangannya saja menggunakan IC LM 7805, 7806, 7809,7812 dan 7824. IC LM tersebut masing – masing akan mengeluarkan tegangan yang stabil diantaranya 5 volt, 6 volt, 9 volt, 12 volt dan 24 volt. Tegangan tersebut digunakan untuk mensuplay sumber daya dari semua pariferal- pariferal dari komponen robot pemadam api

Kebutuhan suplai dari catu daya

- 1 5 volt : sensor ping
- 2 9 volt :sensor UVtron
- 3 12 volt :mikrokontroler, driver motor Washer
- 4 24 volt : driver motor DC L298D

Rangkaian catu daya robot pemadam api



Gambar 3.4 Catu Daya

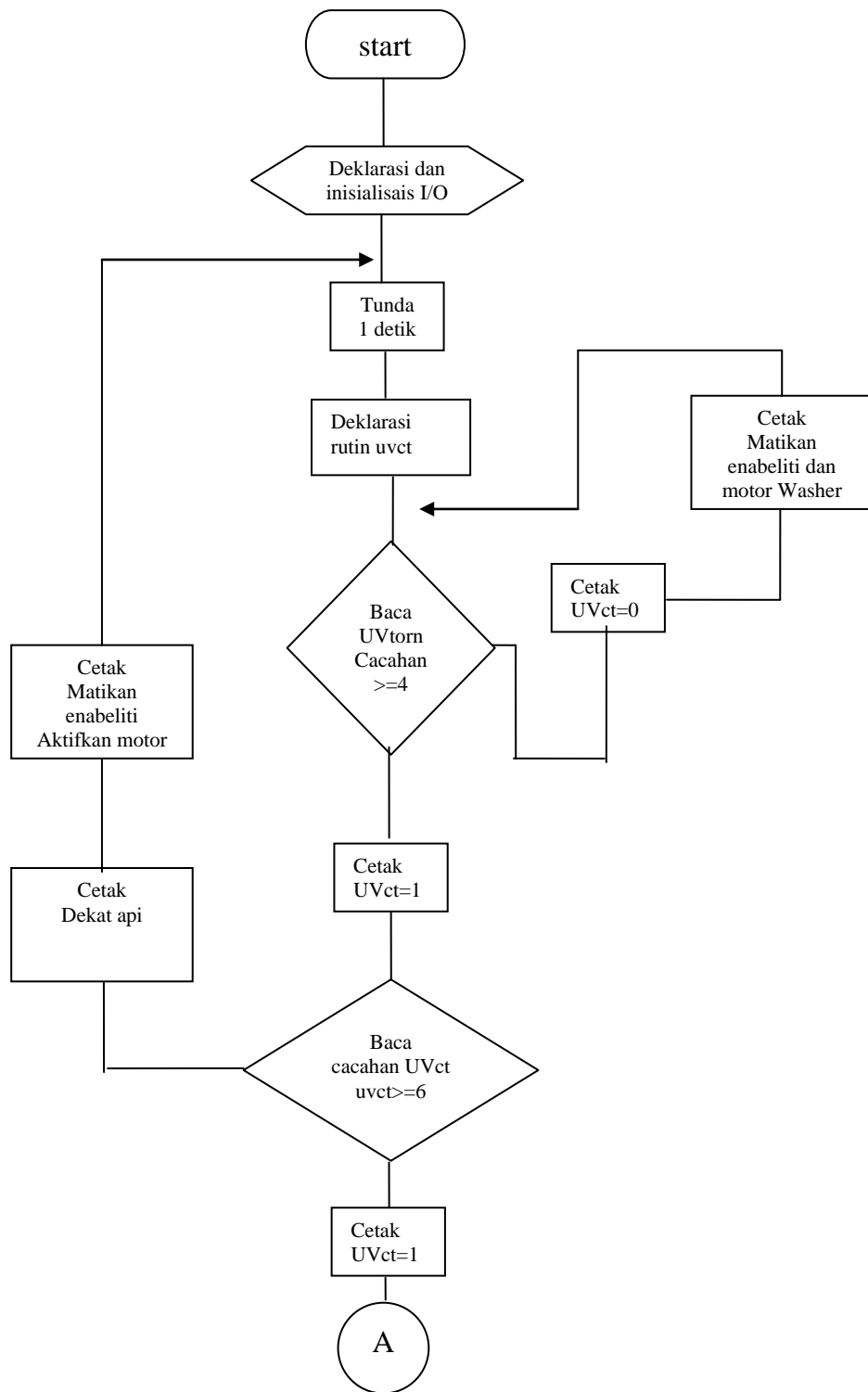
### **3.2 Cara Kerja Sistem Keseluruhan**

Inti dari cara kerja sistem ini seperti terlihat pada blok diagram di atas adalah robot berjalan mencari nyala api ketika telah menemukan nyala api tersebut robot berhenti dan mematikan nyala api tersebut.

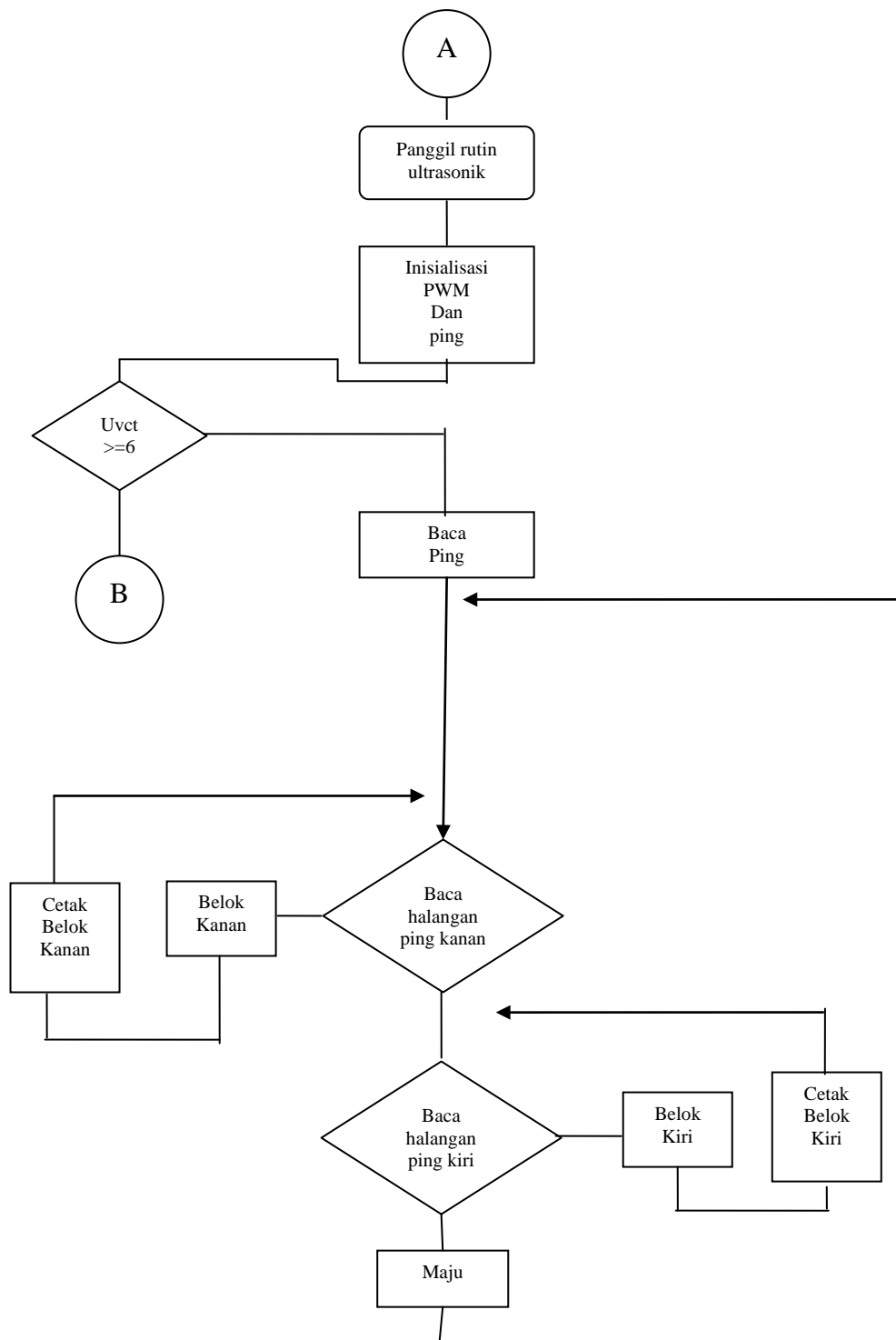
Cara kerja sistem secara keseluruhan seperti berikut:

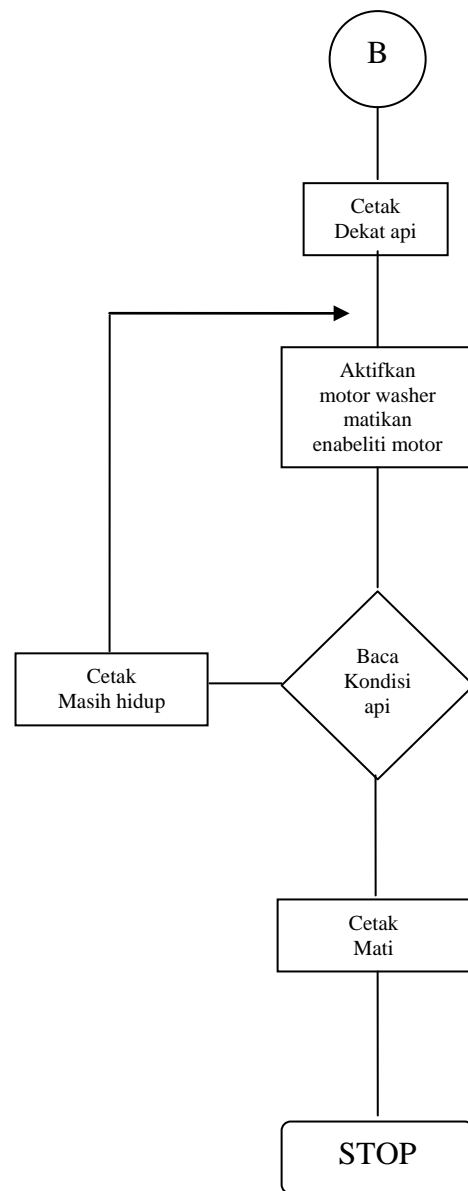
Mikrokontroller ATmega 8535 adalah sebagai otak dari seluruh kinerja robot dalam memadamkan api. Terdiri dari 2 buah mikrokontroler. Mikrokontroler 1 mengontrol enabeliti pada motor, motor washer dan sensor UVtron. Mikrokontroler 2 mengatur motor DC dan sensor Ping. Pertama robot diletakkan pada posisi stand by dengan belum mengaktifkan Switch pada robot, ketika Switch kita posisikan pada posisi On maka sensor UVtron akan mendeteksi adanya nyala api, jika tidak terdapat nyala api maka enabeliti pada robot akan high mengakibatkan motor Dc bekerja membuat robot berjalan untuk mencari nyala api. Dalam proses pencarian nyala api robot dikendalikan oleh Sensor Ultrasonik Ping. Sensor Ultrasonik Ping ini berfungsi sebagai alat navigasi pada robot, dimana sensor ping ini bekerja untuk mendeteksi adanya halangan yang berada di depan, kiri maupun kanan. Jika terdapat halangan didepan maka robot akan mundur, jika terdapat halangan di samping kiri robot maka robot akan berbelok ke kanan dan jika terdapat halangan di samping kanan maka robot akan berbelok ke kiri. Dengan adanya sensor Ultrasonik ini robot tidak akan menabrak halangan yang menghambat laju robot sehingga robot dapat terus berjalan mencari nyala api.

Ketika robot telah menemukan nyala api maka cacahan dari sensor UVtron akan lebih tinggi dari 4 mengakibatkan logika dari sensor UVtron adalah 1, hal tersebut membuat mikrokontroler memberikan logika low pada enabiliti motor DC yang menjadikan motor akan berhenti. Setelah motor DC berhenti dengan delay sebesar 5 detik mikrokontroler langsung mentrigger driver motor Washer menjadikan motor Washer bekerja untuk menyedot air dan menyemburkannya ke nyala api yang bertujuan agar nyala api itu padam. Setelah nyala api padam mikrokontroler langsung memberi logika low pada driver motor Washer, sehingga memaksa motor Washer untuk berhenti bekerja. Dengan delay yang diberikan sebesar 5 detik maka sistem akan kembali dari awal dan sensor UVtron kembali melakukan deteksi api begitu seterusnya.









Gambar 3.5 Diagram Alur Sistem

## **BAB IV**

### **PEMBUATAN BENDA KERJA**

Hasil rancangan yang telah dibuat, selanjutnya direalisasikan dalam bentuk benda kerja yang siap dioperasikan. Proses pembuatan benda kerja dari proyek akhir ini meliputi tiga bagian, yaitu :

- a. Pembuatan bagian elektronik.
- b. Pembuatan bagian mekanik.
- c. Pembuatan program.

#### **4.1 Alat dan bahan yang digunakan**

Dalam pembuatan benda kerja pada tugas akhir ini secara keseluruhan digunakan alat-alat dan bahan sebagai berikut :

##### **4.1.1 Daftar Alat dan Bahan**

Alat-alat yang diperlukan dalam keseluruhan pembuatan benda kerja yaitu:

*Tabel 4-1 Alat Yang Digunakan*

<b>ALAT YANG DIGUNAKAN</b>	<b>SPESIFIKASI</b>	<b>JUMLAH</b>
Spidol tahan air	Snowman	2
Penggaris	30 cm	2
Cutter	Kenko	1 set
Palu	0,5 kg	1
Obeng	(+) dan (-)	2
Mesin bor	60W	1
Mesin bor kecil	Bor PCb Mini drill	1
Mata bor berukuran	0,8 ,1,2 ,3(mm),1cm	1
Gergaji	Besi	1
Kikir	10 cm	1
Penyedot timah	Rayden Gs 300	1

ALAT YANG DIGUNAKAN	SPESIFIKASI	JUMLAH
Tang	Kombinasi	1
Multimeter	Analog	1
PSA	10 A	1
Kunci Pas	8,12	1
Lem tembak	Sanya	1
Setrika	Airlux	1

#### 4.1.2 Daftar bahan

Bahan-bahan yang diperlukan dalam keseluruhan pembuatan benda kerja yaitu :

*Tabel 4-2 Tabel Modul Mikro*

MODUL MIKRO	SPESIFIKASI	JUMLAH
IC	ATMaga 8535	2
Tahanan	10K $\Omega$	2
Soket IC	40 Pin	2
Elco	10 $\mu$ F	2
Crystal	12MHZ	2
Kapasitor	33 pF	2
IC LM 7805		2

*Tabel 4-3 Tabel Modul Catu Daya*

MODUL PSA	SPESIFIKASI	JUMLAH
IC	LM7805	1
IC	LM7809	1
IC	LM78012	1
IC	LM 7824	1
Dioda	N4002	7
Elco	1000 $\mu$ F	6
Elco	470 $\mu$ F	5
Elco	220 $\mu$ F	4
kapasitor	2n10j	6
Tahanan	1K $\Omega$	7
Led		7
Switch	on / off	7
holder fuse		7
fuse	2 A	4
fuse	1A	3

*Tabel 4-4 Tabel Modul Motor DC Driver*

<b>MODUL MOTOR DRIVER</b>	<b>SPESIFIKASI</b>	<b>JUMLAH</b>
Tahanan	1 K $\Omega$	12
Soket IC	L298D	4
Dioda	1AN4002	10
Led		10
Holder fuse		1
Fuse	2 A	1
Swicth	On / Off	1
Motor DC	24V 1,75A	2

*Tabel 4-5 Tabel Modul Motor Washer dan kipas DC*

<b>MODUL KOMPARATOR</b>	<b>SPESIFIKASI</b>	<b>JUMLAH</b>
Led	sedang	2
Led	kecil	1
Tahanan	470 $\Omega$	4
Dioda	1AN4002	3
Holder Fuse		2
Swicth	On / Off	1
fuse	1 A	1
kipas DC	12 V / 0,5 A	2
Motor Washer	12 V / 0,8 A	1

*Tabel 4-6 Tabel Bahan Mekanik*

<b>BAHAN MEKANIK</b>	<b>SPESIFIKASI</b>	<b>JUMLAH</b>
Arkelik	1 m x 1 m	1
spacer		25
Mur baut	3 x 10 mm	29
Mur baut	8 x 10 mm	8
Solasi hitam		1,5 m
Selang	kecil	30 cm
Botol tangki air		1

## **4.2 Bagian Elektronik**

Pembuatan bagian elektronik meliputi :

- Perencanaan rangkaian
- Proses pembuatan papan rangkaian tercetak (*Printed Circuit Board - PCB*)
- Pemasangan komponen

### **4.2.1 Perencanaan Rangkaian**

Perencanaan rangkaian digunakan untuk mendapatkan hasil rangkaian yang optimal sesuai dengan yang diinginkan. Dalam perencanaan ini langkah-langkah yang dilakukan meliputi ukuran papan rangkaian tercetak yang digunakan, jalur rangkaian yang akan dibuat, skema rangkaian dan komponen yang akan digunakan.

### **4.2.2 Pembuatan Papan Rangkaian Tercetak (PCB)**

Proses pembuatan papan rangkaian tercetak meliputi beberapa langkah pengerjaan agar hasil yang didapat bisa lebih baik urutan pembuatan papan rangkaian tercetak yaitu :

- Pembuatan jalur rangkaian pada PCB.
- Penyablonan gambar rangkaian ke PCB.
- Pelarutan.
- Pengeboran.

#### 4.2.2.1 Pembuatan Jalur

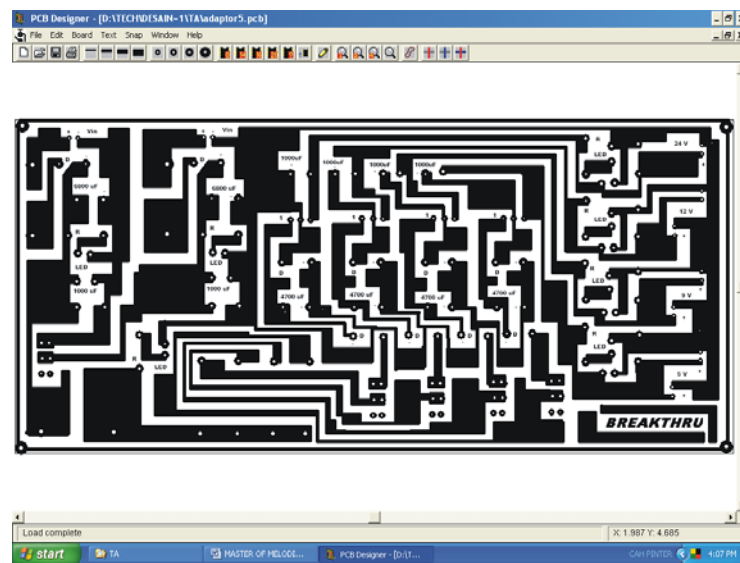
Jalur papan rangkaian tercetak dapat dibuat secara manual maupun disablon. Jika pembuatan dilakukan secara manual, maka pembuatan jalur pada papan rangkaian tercetak hanya dilakukan dengan menggunakan spidol permanent ataupun elektro set. Jika pembuatan dilakukan dengan cara disablon, maka diperlukan sebuah film positif. Film positif ini merupakan transparansi dari hasil cetakan printer setelah dilakukan perancangan jalur dari komputer (jalur dirancang dengan program *PCB Designer*). Sebelum disablon papan tembaga harus dibersihkan terlebih dahulu dari minyak dan kotoran yang melekat dengan menggunakan bensin dan di amplas terlebih dahulu. Setelah bersih, baru dilakukan penyablonan pada bagian tembaga dari papan rangkaian tercetak.

Dalam pembuatan jalur PCB dengan sablon ada beberapa langkah, yaitu :

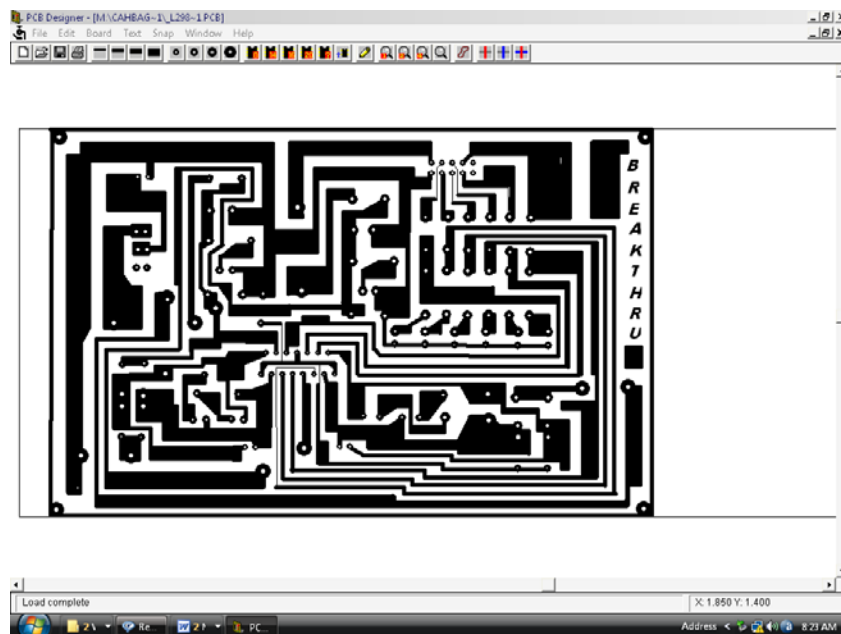
- a. Merancang jalur rangkaian menggunakan program *PCB Designer*, yang kemudian di print pada kertas, lalu di fotokopi dengan menggunakan kertas transparan. Berikut gambar hasil dari pembuatan jalur rangkaian dengan menggunakan PCB.



*Gambar 4.1 proses pembuatan jalur rangkaian dengan PCB designer*

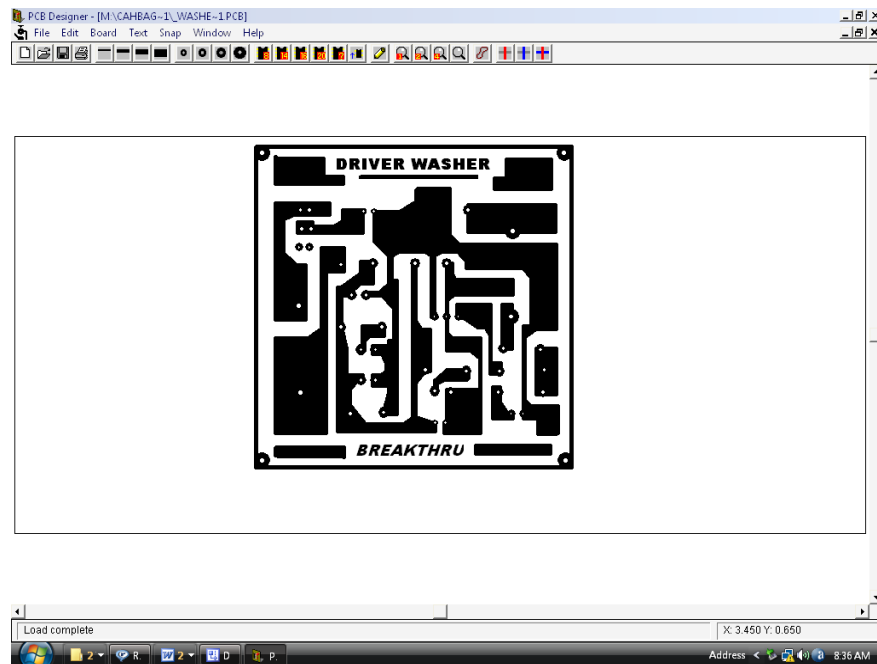


Gambar 4.2 Gambar jalur PCB catu daya

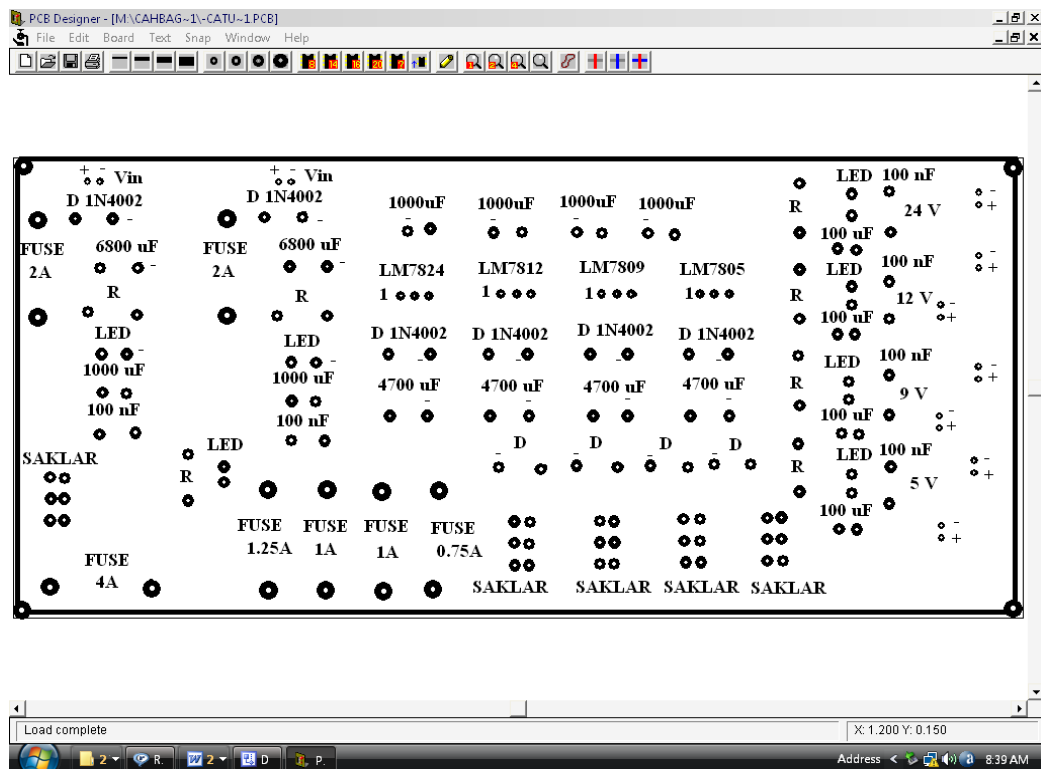


Gambar 4.3 Gambar jalur PCB driver motor DC

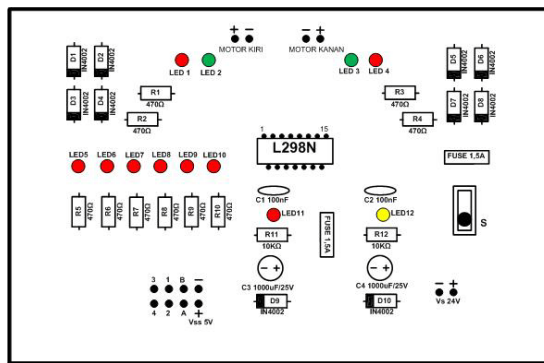




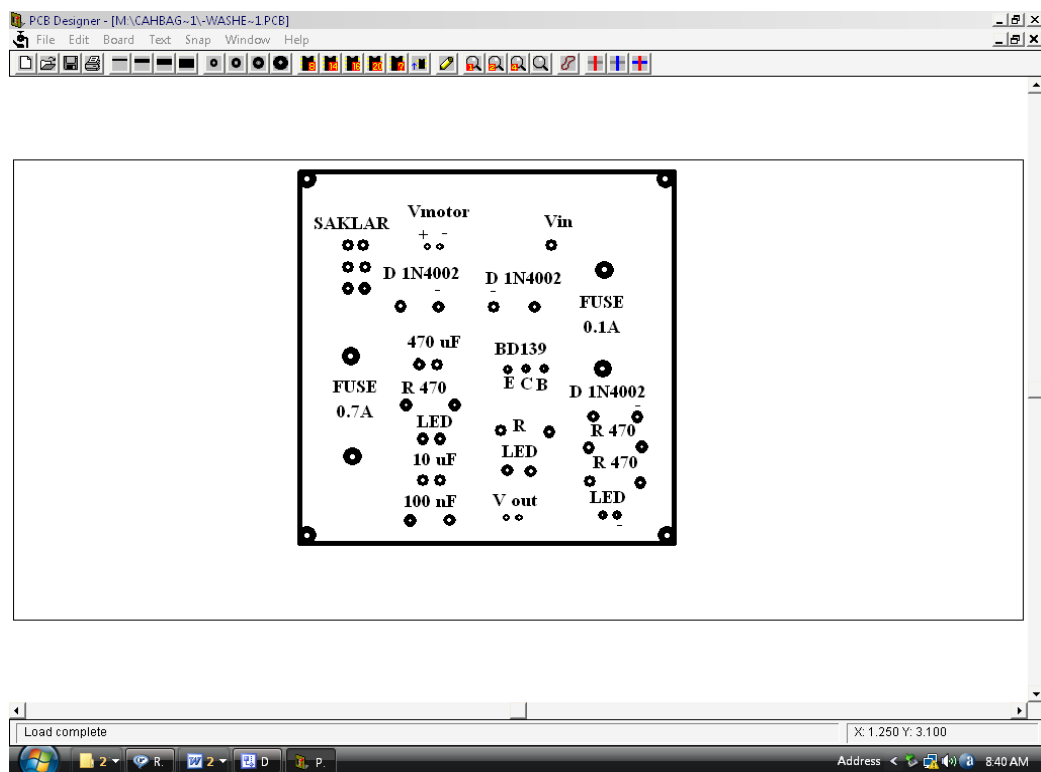
Gambar 4.4 Gambar jalur PCB driver motor washer



Gambar 4.5 Gambar layout tata letak komponen catu daya secara negatif



Gambar 4.6 Gambar layout tata letak komponen driver l298 secara negative



Gambar 4.7 Gambar layout tata letak komponen driver washer secara negative

- b. Memotong PCB polos sesuai dengan ukuran yang dibutuhkan, karena benda kerja yang dibuat terdiri dari beberapa modul rangkaian.
- c. Memindahkan jalur yang telah dicetak ke permukaan PCB dengan cara di seterika.

- d. Membersihkan permukaan tembaga agar bebas dari kotoran atau minyak yang dapat menghalangi proses pelarutan.

#### 4.2.2.2 Proses Pelarutan

Untuk proses pelarutan dilakukan langkah-langkah sebagai berikut :

- Melarutkan papan PCB yang telah tercetak jalur rangkaian dengan larutan ferichloride ( $\text{FeCl}_3$ ) untuk menghilangkan tembaga dari papan rangkaian tercetak yang tidak terpakai. Agar proses pelarutan dapat berjalan lebih cepat maka diperlukan larutan *ferichloride* yang pekat, dan menggunakan air panas sebagai pelarutnya.
- Mengangkat papan rangkaian tercetak dan mencucinya dengan air setelah lapisan tembaga yang tidak terpakai hilang.
- Membersihkan sisa air dengan lap yang kering dan bersih.
- Membersihkan sisa lapisan pada jalur (bekas spidol / cat sablon) dengan bensin.

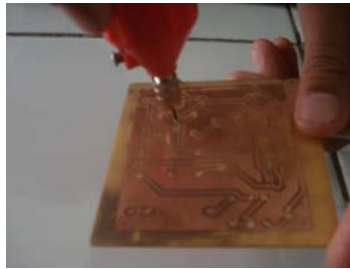


Gambar 4.8 proses pelarutan PCB

#### 4.2.2.3 Proses Pengeboran

Setelah proses pelarutan selesai, selanjutnya dilakukan pengeboran pada titik-titik pengeboran yang telah ditentukan untuk memasang komponen. Pada

proses pengeboran digunakan mata bor yang berdiameter 0.8 mm dan 1 mm untuk lubang komponen dan mata bor berdiameter 3 mm untuk lubang tepi sebagai tempat baut. Ukuran diameter berdasarkan kaki komponen yang akan dipasang. Untuk lubang-lubang yang mempunyai ukuran lain digunakan mata bor yang diameternya sesuai.



*Gambar 4.9 proses pengeboran PCB*

#### **4.2.3 Proses Pemasangan Komponen**

Sebelum komponen dipasang pada papan rangkaian tercetak, kaki-kaki komponen perlu dibersihkan dahulu agar tidak ada kotoran yang menempel, karena dapat mengganggu hasil penyolderan. Langkah-langkahnya sebagai berikut:

- Mengolesi jalur papan rangkaian tercetak dengan bensin agar dalam proses penyolderan timah lebih mudah melekat.
- Memasang komponen sesuai pada tempatnya kemudian menyoldernya dan memotong sisi kaki komponen dengan gunting.
- Membersihkan papan dari sisa-sisa penyolderan dengan bensin dan memeriksa jalur rangkaian.



*Gambar 4.10 Proses pemasangan komponen*

Perlu diperhatikan sebelumnya komponen-komponen yang akan dipasang harus diperiksa dulu kondisi fisiknya apakah komponen tersebut dalam kondisi baik atau sudah rusak. Komponen yang rusak atau sudah tidak sesuai dengan karakteristiknya harus diganti untuk menghindari kegagalan di dalam pengoperasiannya. Gunakan solder dengan daya yang tidak terlalu besar yaitu sekitar 30 watt. Hal ini untuk menghindari terjadinya pemanasan yang berlebihan terhadap komponen-komponen. Pada saat penyolderan komponen, disarankan jangan terlalu lama, yaitu sekitar 3.4 detik agar komponen yang disolder tidak rusak karena terlalu panas. Timah solder menggunakan jenis yang kualitasnya baik guna mendapatkan hasil penyolderan yang baik.

### **4.3 Bagian Mekanik**

Proses pembuatan mekanik meliputi proses pembuatan bentuk benda kerja yang berupa body robot. Pembuatan benda kerja ini memerlukan beberapa langkah kerja agar didapat hasil sesuai yang diharapkan.

#### **4.3.1 Proses Pembuatan Benda Kerja**

Proses pembuatan benda kerja dalam tugas akhir ini meliputi :

- Menggambar bentuk alat yang akan dibuat pada arkelik

- Menandai arkelik sesuai dengan bentuk dan ukuran yang direncanakan..
- Memotong bagian yang sudah ditentukan dengan alat pemotong (gergaji atau pemotong plat).
- Mengebor arkelik yang telah diberi tanda.
- Merangkai arkelik dengan mur baut atau dengan mengencangkan sesuai dengan gambar rancangan.
- Mengamplas dan mengecat alat.

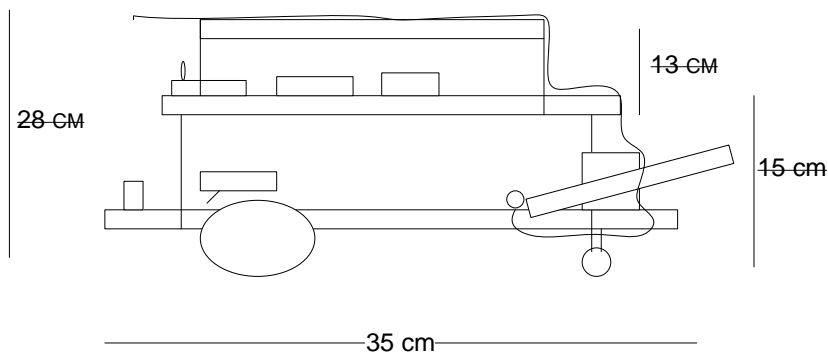
#### **4.3.2 Proses Perakitan**

Proses perakitan rangkaian pada alat ini meliputi :

- Merakit bagian dalam benda kerja yaitu tempat menempelnya papan rangkaian tercetak dengan menggunakan penyangga ( Spacer ).
- Menghubungkan papan rangkaian tercetak satu dengan yang lainnya dengan cara memasang kabel antar papan rangkaian tercetak.
- Memasang baut yang diperlukan pada bagian yang telah diberi tanda atau lubang.



*Gambar 4.11 gambar proses perakitan*



*Gambar 4.12 Gambar Skema fire fighting robot*

Cara pembuatan body robot:

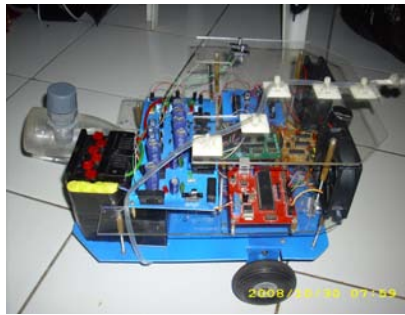
Body robot dibuat dari arkelik. Arkelik diukur dan digambar skemanya untuk dipotong dengan gergaji. Setelah dipotong sesuai dengan ukurannya kemudian arkelik dibor untuk tempat mur dan baut rangkaian robot setelah jadi arkelik di clear terlebih dahulu sebelum di pilox. Gambar 4.9 merupakan gambar body robot tampak nyata..



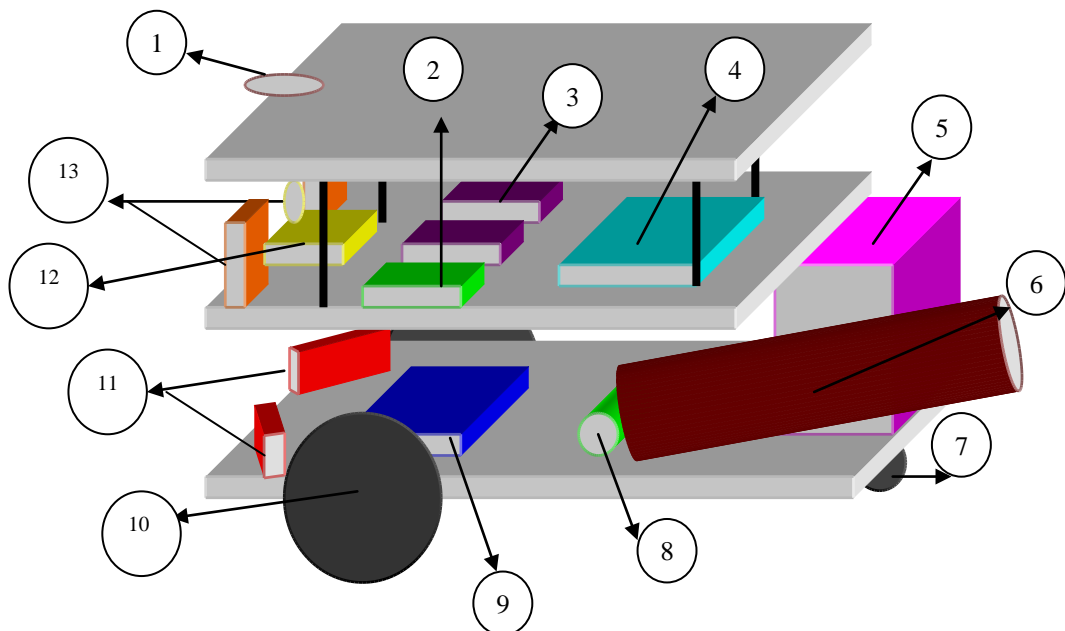
*Gambar 4.13 Gambar Nyata bagian lantai 1 robot*



*Gambar 4.14 Gambar Nyata robot tampak samping*



Gambar 4.15 Gambar Nyata finising robot tampak samping



Gambar 4.16 Gambar Dimensi robot

NOMOR	KETERANGAN	DIMENSI
1	Pemadam api dengan air	diameter = 0,5 cm
2	Driver motor <i>washer</i>	8,5 x 8 cm
3	Sistem minimum mikroprosesor ATmega8535	7,5 x 6,5 cm
4	Rangkaian catu daya	21 x 10 cm
5	Accumulator/baterai	7 x 8 cm, 9,5 x 4,5 cm
6	Tandon air	21 x 3 cm



7	Roda castor	diameter = 1 cm
8	Motor washer	5 x 1 cm
9	Driver motor DC IC L298N	16 x 10 cm
10	Roda <i>FFR</i> (motor DC)	diameter = 6 cm
11	Sensor Ultrasonik	4,5 x 2 cm
12	Sensor UVTron	6 x 6 cm
13	Kipas pemadam api	7,5 x 7,5 cm

#### 4.4 Proses Pembuatan Program

Perangkat lunak yang digunakan pada tugas akhir ini ditulis pada teks editor dalam bahasa *C* dan software yang digunakan adalah Code Vision Avr. setelah program selesai dibuat, kemudian program disimpan dengan nama file yang berekstensi *.C*. Selanjutnya program yang telah selesai dibuat tadi dilakukan *compiling* dari bahasa *C* ke dalam kode-kode instruksi mesin yang sesuai dengan *up-code* mikrokontroler Intel dengan berekstensi *HEX*.

Dalam pembuatan tugas akhir ini penggunaan perangkat lunak sangat penting, mengingat perangkat lunak digunakan untuk pengaturan dari keseluruhan kerja sistem baik perangkat keras maupun perangkat lunak itu sendiri. Langkah-langkah pembuatan program tersebut adalah sebagai berikut :

- Membuat diagram alir (*flow chart*) dari program yang akan dibuat.
- Membuat program menggunakan pemrograman *assembler* dengan referensi diagram alir.
- Mengkompilasi program yang telah dibuat sampai tidak terjadi kesalahan.
- Pengisian program.

#### **4.4.1 Pembuatan Diagram Alir**

Dalam menyusun diagram alir diusahakan dapat membagi proses yang kompleks menjadi sub program yang lebih kecil, sehingga pencarian kesalahan akan lebih mudah. Selain itu akan memudahkan orang lain dalam membaca alir program yang dibuat.

#### **4.4.2 Pembuatan Program**

Penulisan program dilaksanakan setelah diagram alir selesai dirancang. Pemilihan editor *teks* disesuaikan dengan kebiasaan dan kesenangan. Agar *teks* yang telah dibuat nantinya dapat dimengerti oleh program AVR prog maka hasil penulisan program harus dibuat dengan ekstension *inc* dan dirubah menjadi ekstensi *hex*..

#### **4.4.3 Kompilasi Program**

Program yang ditulis menggunakan editor *teks* kemudian dikompilasi dengan menggunakan program Code vision AVR. Bila tidak ada pesan kesalahan, proses kompilasi telah berhasil. Bila ada pesan kesalahan, dapat dicari kesalahan yang terjadi berdasarkan informasi pesan kesalahan tersebut.

#### **4.4.4 Pengisian Program**

Perlengkapan yang dibutuhkan dalam pengisian *flash* ATmega 8535, diantaranya :

- Catu Daya 9 V.
- Kabel isp kanda STK 200 *using paralel progaming*.

- PC dengan Sistem Operasi *MS Windows 9X/me/NT/2000/XP*.
- Perangkat lunak pemrograman ( AVR prog )

Pengisian program yang telah dikompilasi dalam format heksadesimal (hex), diisikan ke dalam *Flash EPROM* ATMega 8535 dengan menjalankan perangkat lunak pemrogram.

Urutan cara pengisian program ke dalam *Flash EPROM* ATMega 8535 adalah sebagai berikut :

- Pastikan kabel STK kanda 200 antara PC dengan rangkaian *programmer* ATMega 8535 telah terpasang.
- Pasang IC ATMega 8535 pada soket yang telah ditentukan pada rangkaian pengisi *Flash EPROM Programmer* ATMega 8535.
- Berikan catu tegangan DC 9 V ke rangkaian pengisi *Flash EPROM Programme* ATMega 8535 .
- Jalankan perangkat lunak pemrograman.

## **BAB V**

### **PENGUKURAN DAN PENGUJIAN**

#### **5.1. Pengujian Alat**

Sebelum dilakukan pengoperasian terhadap benda kerja yang telah dibuat, untuk mengetahui keberhasilan dan kelayakan benda kerja maka terlebih dahulu dilakukan pengujian terhadap benda kerja.

##### **5.1.1. Tujuan**

Pengujian berguna untuk menghasilkan suatu data-data spesifikasi alat atau untuk mendapatkan titik-titik pengukuran pada alat yang dibuat, sehingga pada saat terjadi kesalahan atau kerusakan dapat dianalisa dengan tepat dan mudah dalam perbaikannya.

Tujuan dari pengujian alat adalah sebagai berikut:

1. Memastikan bahwa alat tersebut dapat bekerja dengan baik.
2. Mengetahui kondisi sinyal alat tersebut pada titik-titik tertentu.
3. Mengetahui harga tegangan pada titik tertentu, apakah sudah sesuai dengan harga sewajarnya.
4. Menganalisa kerja alat secara lebih terperinci.

##### **5.1.2. Alat dan Bahan yang Digunakan**

Untuk memudahkan dalam pengujian dan mendapatkan data-data yang akurat, maka alat dan bahan yang dipergunakan adalah:

1. Multimeter analog.

2. Catu daya 5V dan 9V.
3. Kabel penghubung.

### **5.1.3. Langkah-langkah Umum**

Langkah-langkah umum yang harus dilakukan pada pengujian adalah:

1. Mempersiapkan benda kerja dan perlengkapannya.
2. Mempersiapkan gambar rangkaian dan tata letak komponen.
3. Mempersiapkan semua peralatan dan bahan yang akan digunakan.
4. Melakukan pengujian.
5. Mengukur tegangan dan arus pada titik-titik tertentu.

### **5.1.4. Pengujian Rangkaian Catu Daya**

Pengujian catu daya bertujuan untuk mengetahui tegangan pada keluaran IC LM 7809, IC LM 7812 dan IC LM 7824. Langkah-langkah pengujian pada catu daya adalah sebagai berikut:

1. Memutuskan jalur keluaran yang menuju rangkaian aplikasi dan sistem mikrokontroler.
2. Menghidupkan catu daya.
3. Mengukur tegangan pada titik titik yang telah ditentukan .

Hasil pengukuran pada rangkaian catu daya

*Tabel 5.1 Tegangan Output Rangkaian Catudaya*

No	Keterangan	Tegangan
1	Akumulator	12 V
2	Akumulator	12 V
3	Output IC 7809	8,7 V
3	Output IC 7812	11,6 V
4	Output IC 7824	24,3 V

#### **5.1.5. Pengujian Sensor Ultrasonik**

Langkah pengujian sensor Ultrasonik adalah sebagai berikut:

1. Menghubungkan catu daya 5V pada sigal SIG sensor Ultrasonik.
2. Menghidupkan rangkaian mikrokontroler yang telah diprogram untuk menguji sensor ultrasonik dan mematikan rangkaian aplikasi yang lain.
3. Mengukur jarak deteksi sensor pada jara antara objek dengan sensor ultrasonik saat sensor ultrasonik terhalang atau tidak terhalang.
4. melakukan pengujian dengan media yang berbeda untuk mengetahui kualitas sensor tersebut.

*Tabel 5.2 pengukuran jarak sensor ultrasonik*

No	Media yang digunakan ( objek )	sensor kanan		sensor kiri	
		kisaran maksimal	kisaran minimum	kisaran maksimal	kisaran minimum
1	Dinding	256 cm	4,3 cm	235 cm	3,5 cm
2	Papan / Kayu	185 cm	4,2 cm	195 cm	2,9 cm
3	Steroform	tidak dipantulkan	tidak dipantulkan	tidak dipantulkan	tidak dipantulkan
4	plastik	tidak dipantulkan	tidak dipantulkan	tidak dipantulkan	tidak dipantulkan

Data – data hasil pengukuran jarak sensor ultrasonik digunakan un untuk proses kinerja robot. Data jara dari sensor ultrasonik akan digunakan untuk acuan bagi robot dalam menentukan arah belokan, sehingga robot tidak dapat menabrak saat terdapat halangan yang menghambat laju robot tersebut.



*Gambar 5.1 pengujian sensor ultrasonik terhadap dinding*



Gambar 5.2 pengujian sensor ultrasonik terhadap steroform

#### 5.1.6. Pengujian Rangkaian Sistem Mikrokontroler

Langkah-langkah pengujian pada rangkaian sistem Mikrokontroler adalah sebagai berikut:

1. Lepaskan semua hubungan dengan rangkaian aplikasi.
2. Pasang IC mikrokontroler yang telah diisi program untuk menyalakan semua port.
3. Menghidupkan sistem mikrokontroler dalam keadaan tanpa masukan dan keluaran dengan tegangan 5 V.

Dari hasil pengukuran diperoleh data-data sebagai berikut:

*Tabel 5.3 Tegangan Logika Port Mikrokontroler ATmega 8535*

Port	H / L	Tegangan pada masing-masing pin (V)							
		.0	.1	.2	.3	.4	.5	.6	.7
A	H	4.90	4.92	4.92	4.90	4.90	4.92	4.92	4.90
	L	0.04	0.05	0.05	0.04	0.05	0.04	0.04	0.04
B	H	4.62	4.60	4.62	4.62	4.60	4.60	4.60	4.62
	L	0.02	0.03	0.03	0.03	0.03	0.02	0.03	0.02
C	H	4.58	4.57	4.58	4.58	4.57	4.59	4.57	4.56



	L	0.01	0.02	0.01	0.01	0.02	0.01	0.01	0.01
D	H	4.62	4.60	46.1	4.61	4.60	4.62	4.60	4.60
	L	0.02	0.02	0.03	0.01	0.02	0.02	0.03	0.02

### 5.1.7 Pengujian Pada sakelar – sakelar ( limith Swicth )

Pada pengujian *Limit Switch*, diukur berapa tegangan saat *Limit Switch* dalam keadaan tertutup (terhubung dengan Vcc) dan keadaan pada saat *Limit switch* terhubung dengan negatif.

*Tabel 5.4 Keadaan Limit switch*

No	Sakelar VCC 1	Sakelar VCC 2
1	saat on = 12 V	saat on = 12 V
2	saat off = 0 V	saat off = 0 V
3	Saat Seri = 24 V	

### 5.1.8. Pengujian Sensor Uvtron

Langkah pengujian sensor Ultrasonik adalah sebagai berikut:

- 1 Menghubungkan catu daya 8V pada PIN Q SENSOR UVTron.
- 2 Menghidupkan rangkaian mikrokontroler yang telah diprogram untuk dapat mengaktifkan sensor UVTron
- 3 Melakukan pengujian dengan memberikan nyala api pada robot.



Gambar 5.3 pengujian sensor UVTron terhadap nyala api.

## 5.2 Hasil Dari Pengujian Alat

Setelah melakukan pengujian dan pengukuran, ada beberapa hasil yang berkenaan dengan bekerjanya alat yaitu:

1. Tegangan logika 1 pada keseluruhan sistem merupakan tegangan Vcc yang sudah mengalami banyak hambatan didalam serpih sehingga tegangannya menurun.
2. Saat diumpankan pada masukan rangkaian, tegangan logika 1 mengalami penurunan akibat adanya impedansi masukan rangkaian.
3. Tegangan Vcc saat aplikasi bekerja terukur lebih rendah akibat tegangan jatuh oleh beban.
4. pembacaan sensor – sensor pada alat sering berubah – ubah.
5. Ada beberapa hal lain yang mempengaruhi kerja dari rangkaian yaitu :
  - A. Hubungan jalur PCB yang kurang tebal atau putus.
  - B. Kebersihan papan rangkaian dari lemak solder.
  - C. Hubungan jalur *ground* yang kurang sempurna.

D. Kondisi komponen yang kurang baik.

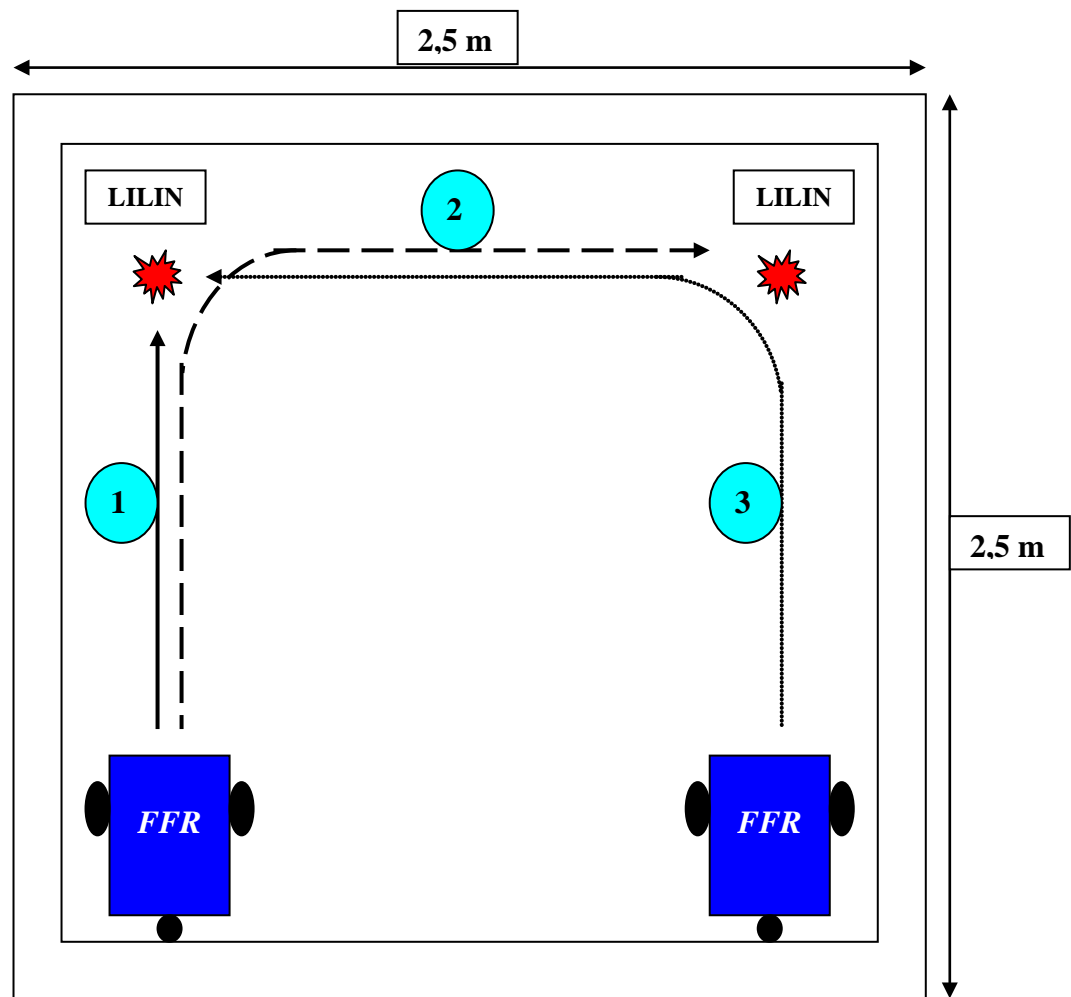
Dibawah ini merupakan data dari hasil pengujian pada robot pemadam api berbasis mikrokontroler ATmega 8535.

*Tabel 5.5 Pengujian alat*

No	lokasi pengujian	waktu
1	BULUSAN	jam 10.00
2	TLOGOSARI	jam 09.00
3	Ruang HME T elektro	jam 13.00



*Gambar 5.4 gambar robot telah memadamkan api*



*Gambar 5.5 Gambar skema pengujian robot*

Berdasarkan hasil pengukuran yang dilakukan pada tanggal 28 November 2008 dapat diambil kesimpulan bahwa, batas ukur kemampuan sensor ultrasonik dalam mendeteksi objek adalah 256 cm akan tetapi dalam pemasangan pada robot pemadam api deteksi sensor ultrasonik disetting 15 cm dalam mendeteksi objek data hasil pembacaan sensor akan menentukan arah belokan dari robot ini. Agar sistem kerja robot dapat berjalan dengan baik maka settingan sensor UVTron adalah berbeda dengan sensor ultrasonik hal ini diperuntukkan agar trobot dapat mendeteksi adanya nyala api sebelum sensor ultrasonik memerintahkan robot untuk berbelok. Jadi robot dapat bekerja memadamkan nyala api.

### 5.3. Spesifikasi Benda kerja

Spesifikasi dari benda hasil Tugas Akhir ini adalah sebagai berikut:

1. Nama alat : Robot Pemadam Api Berbasis Mikrokontroler  
ATMega 8535
2. Mikrokontroler : ATMega 8535
3. Memori program : *Flash 8Kb internal*
4. Kristal osilator : 12 MHz
5. Bahasa pemrograman : C
6. Program bantu : code vision AVR
7. Catu daya rangkaian : + 9 Volt, +5 Volt,
8. Sensor – sensor : Sensor jarak ultrasonik, Sensor flame UVTron
9. Body robot : Arkelik
10. Berat dan pajang robot: 4,3 Kg / 35 cm

### 5.4. Petunjuk Pengoperasian

Dalarn pengoperasian diperlukan suatu langkah-langkah dan perlengkapan yang diperlukan agar alat dapat bekerja dengan semestinya dan bagi pengguna tidak mengalami banyak kesulitan. Untuk melakukan pengoperasian benda kerja secara umum dapat dilakukan langkah-langkah sebagai berikut:

1. Menyalakan benda kerja dengan menekan sakelar utama.
2. tekan semua sakelar catu daya agar catu seluruh rangkaian tersuplay tegangannya.
3. menyalakan sakelar pada masing – masing sensor

4. Apabila lampu indikator menyala hal tersebut menandakan bahwa seluruh sensor dalam keadaan sedang bekerja.
5. Apabila lampu indikator mati berarti sensor tidak bekerja, segera cek pada sekering dalam *fuseholcing* apakah putus atau tidak. Jika putus gantilah dengan yang baru.
6. Saat mendeteksi objek atau dekat dengan objek maka robot akan berbelok jika tidak berbelok cek pada catu daya sensor ultrasonik.
7. Saat mendeteksi adanya nyala api maka motor akan berhenti dan kipas akan menyala untuk mematikan nyala api, jika kipas tidak nyala cek indikator sensor UVTron nyala atau tidak jika nyala maka cek pada sakelar driver motor washer
8. ketika robot telah berhasil mematikan nyala api maka sistem secara otomatis akan mengeset pada kondisi awal, jika ingin mematikan sensor maka matikan sakelar utama maka robot akan berhenti bekerja.

## **BAB VI**

### **PENUTUP**

#### **6.1 Kesimpulan**

Dengan rahmat dan karunia Tuhan Yang Maha Esa, penulis telah menyelesaikan penyusunan laporan tugas akhir ini. Dari laporan tugas akhir yang berjudul “Sensor Ultrasonik Sebagai Alat Navigasi Robot Pada Robot Pemadam Api Berbasis Mikrokontroler ATmega 8535” ini dapat diambil kesimpulan sebagai berikut :

- 1 Robot pemadam api ini dibuat sebagai gambaran alat simulasi yang berfungsi untuk memudahkan pekerjaan manusia terutama dalam bidang pemadam kebakaran.
2. Robot pemadam api ini dilengkapi dengan sensor ultrasonik yang berfungsi untuk mendeteksi adanya objek yang berada didepan robot sehingga memberikan instruksi kapan robot harus berbelok
3. Pada robot pemadam api ini sensor ultrasonik dapat bekerja mendeteksi objek dengan kisaran maksimal adalah 256 cm dan kisaran minimal pembacaan sensor adalah 3 cm pada dinding tembok sedangkan untuk penggunaan dinding lain seperti kayu, plastik maupun sterofoam kualitas deteksi sensor tidak sebagus dari penggunaan dinding tembok.
4. Pembacaan sensor kiri dan kanan tidak sama hal ini dipengaruhi oleh kualitas masing – masing sensor yang berbeda.
5. Pemrograman sensor dengan bahasa C sangat kompatibel dimana dapat digunakan untuk menganalisa data dari deteksi sensor terhadap objek

yang dideteksi, data ini sangat penting guna memaksimalkan kinerja sensor ultrasonik terhadap sistem

## **6.2 Saran**

Alat yang dibuat penyusun sesungguhnya masih terdapat banyak kelemahan, baik secara fisik maupun sistem kerjanya. Oleh karena itu masih perlu kajian – kajian dan serangkaian uji coba lagi agar diperoleh alat yang lebih sempurna. Saran yang bisa penyusun sampaikan adalah :

1. Pembacaan sensor ultrasonic kiri dan kanan tidak sama oleh karena itu untuk mengantisipasi agar kedua sensor dapat menghasilkan pembacaan jarak yang sama dapat digunakan program yang berbeda.
2. Dalam pembuatan software sebaiknya perlu diperhatikan koordinasi untuk data pembacaan dari sensor ultrasonic dengan sensor UVTron agar dapat menentukan jarak yang tepat untuk memadamkan api sehingga menjadikan sensor ultrasonic dan sensor UVTron dapat bekerja dengan maksimal.
3. Pembuatan robot pemadam api dapat dilegkapi dengan LCD sebagai tampilan untuk memonitoring PWM dan jarak dari data ultrasonik



## DAFTAR PUSTAKA

- Bejo, Agus. *C & AVR Rahasia Kemudahan Bahasa C Dalam Mikrokontroler ATmega 8535*. Yogyakarta : Graha Ilmu, 2008
- Budhiarto, Widodo. 2004. *Interfacing Komputer dan Mikrokontroller*. Jakarta: Elex Media Komputindo
- .....2006. *Belajar Sendiri Membuat Robot Cerdas*. Jakarta: Elex Media Komputindo
- .....2007. *12 Proyek Mikrokontroler Untuk pemula*. Jakarta : Elex Media Komputindo
- .....2008. *10 Proyek Robot Spektakuler*. Jakarta: Elex Media Komputindo
- Budioko, Totok. 2005. *Belajar Dengan Mudah dan Cepat Pemrograman Bahasa C dengan SDCC*. Jakarta : Gava Media
- Datasheet Mikrokontroler ATmega 8535
- Malik, Ibnu. 2003. *Pengantar Membuat Robot*. Yogyakarta : Penerbit Gava Media,
- .  
Haryanto. 2005. *Pemrograman Bahasa C Untuk Mikrokontroler ATmega 8535*. Jakarta : Penerbit. PT. Elex Media Komputindo.
- Sumber – Sumber dari internet ([www.Atmel.com](http://www.Atmel.com)), 16 juni 2008, 12.36
- Tim lab. Mikroprosesor. 2006. Pemrograman Mikrokontroler AT89S51 dengan C/C++. Yogyakarta: Andi
- Bhudiarto, Widodo. 2008. *Work Shop Pemrograman Mikrokontroler ATmega 8535 Dengan Bahasa C. Bekasi*

# LAMPIRAN

1

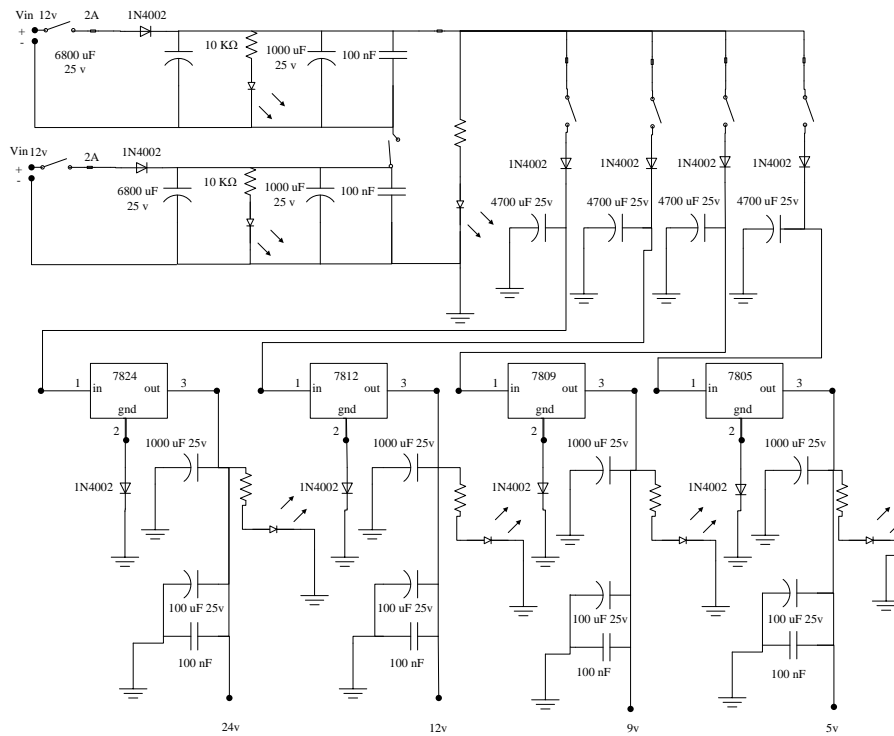
Gambar lay out PCB



# LAMPIRAN

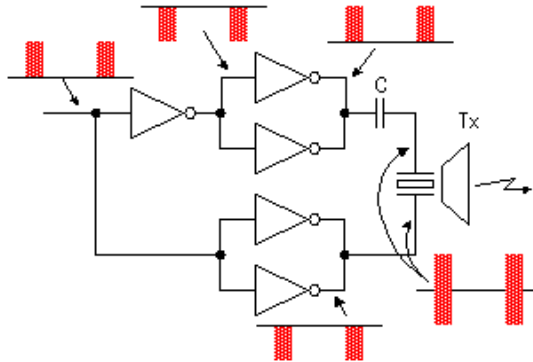


Gambar Rangkaian

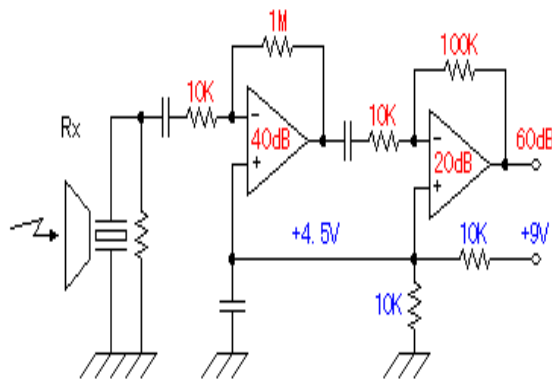


<p align="center"><b>SENSOR ULTRASONIK SEBAGAI ALAT NAVIGASI ROBOT PADA ROBOT PEMADAM API BERBASIS MIKROKONTROLER AVR ATMega 8535</b></p>			
<p><b>SKALA :</b></p>		<p><b>DELTA AGUS</b></p>	
<p><b>TANGGAL :3-11-2008</b></p>		<p><b>LOF 005 464</b></p>	
<p><b>RANGKAIAN CATU DAYA</b></p>		<p><b>ROBOT PEMADAM API</b></p>	
<p align="center"><b>PROGRAM STUDI DIPLOMA III</b></p>		<p align="center"><b>TEKNIK ELEKTRO FAKULTAS TEKNIK</b></p>	
<p align="center"><b>A4</b></p>		<p align="center"><b>A4</b></p>	





Rangkaian Pengendali Sensor Ultrasonik



Rangkaian Sinyal Amplifikasi

**SENSOR ULTRASONIK SEBAGAI ALAT NAVIGASI ROBOT  
PADA ROBOT PEMADAM API  
BERBASIS MIKROKONTROLER AVR ATMega 8535**

SKALA :

DELTA AGUS

Rangkaian pemancar dan penerima

TANGGAL :3-11-2008

LOF 005 464

sensor ultrasonic Ping



**PROGRAM STUDI DIPLOMA III  
TEKNIK ELEKTRO FAKULTAS TEKNIK**

**A4**

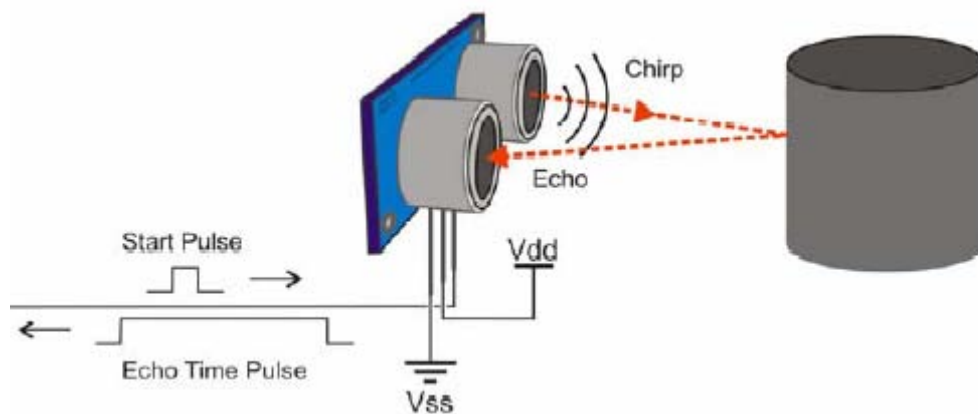
# LAMPIRAN




**Gambar Alat**



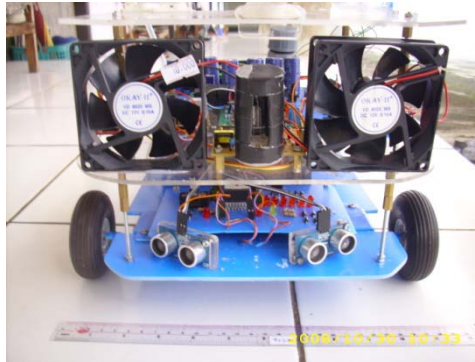
Sensor jarak ultrasonic Ping



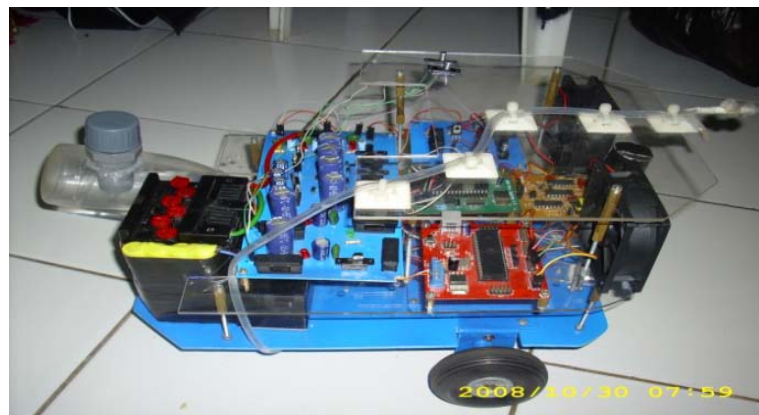
<p>SENSOR ULTRASONIK SEBAGAI ALAT NAVIGASI ROBOT          PADA ROBOT PEMADAM API          BERBASIS MIKROKONTROLER AVR ATMega 8535</p>		
SKALA :	DELTA AGUS	SENSOR JARAK
TANGGAL :3-11-2008	LOF 005 464	ULTRASONIC PING
	<p><b>PROGRAM STUDI DIPLOMA III</b>  <b>TEKNIK ELEKTRO FAKULTAS TEKNIK</b></p>	<p><b>A4</b></p>



	<b>UNIVERSITAS DIPONEGORO SEMARANG</b>	
--	--	--



Robot tampak Depan



Robot tampak Samping

<b>SENSOR ULTRASONIK SEBAGAI ALAT NAVIGASI ROBOT PADA ROBOT PEMADAM API BERBASIS MIKROKONTROLER AVR ATmega 8535</b>		
<b>SKALA :</b>	<b>DELTA AGUS</b>	<b>BENTUK ROBOT</b>
<b>TANGGAL :3-11-2008</b>	<b>LOF 005 464</b>	
	<b>PROGRAM STUDI DIPLOMA III TEKNIK ELEKTRO FAKULTAS TEKNIK</b>	<b>A4</b>

# LAMPIRAN



**Listing Program**

## Listing program mikrokontroler 2

/\*\*\*\*\*\*

This program was produced by the  
CodeWizardAVR V1.24.2 full version  
Automatic Program Generator  
© Copyright 1998-2003 HP InfoTech s.r.l.  
<http://www.hpinfotech.ro>  
e-mail:office@hpinfotech.ro

Project : Robot Pemadam Api ( Fire Figthing Robot )  
Version : Full  
Date : 15/09/2008  
Author : Delta Agus S A  
Company : Delt@\_Ckp\_Software Development  
          Jl Raya Kaligawe II / 2 No 22 Semarang  
Comments : Scaning For Counter UV flame Detecor by delta\_master

Chip type : ATmega8535  
Program type : Application  
Clock frequency : 11.059200 MHz  
Memory model : Small  
External SRAM size : 0  
Data Stack size : 128

\*\*\*\*\*/

```
#include <mega8535.h>
#include <delay.h>
```

```
#define PULSA PORTB.0 // inisialisasi uvtron
#define UVCT PINB.0
#define FLAME DDRB.0
#define OUT 1
#define INPUT 0
```

```
/* prosedur rutin main uvtron*/
```

```
void scan (void)
{
  CACAH++;

  while(UVCT==0){};
  while(UVCT==1)

  if(CACAH>=1){
```

```
PULSA=0;
FLAME=INPUT;
PORTC.2=0;high();}
delay_us(5);
```

```
if(CACAH>4){
FLAME=OUT;
PULSA=1;
PORTC.2=1;low();
delay_ms(10000);
}
}
```

```
void main (void)          // deklarasi utama
{
```

```
// Declare your local variables here
// Input/Output Ports initialization
// Port A initialization
// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In
Func7=In
// State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T
```

```
PORTA=0xFF;
DDRA=0x00;
```

```
// Port B initialization
// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In
Func7=In
// State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T
```

```
PORTB=0x00;
DDRB=0x00;
```

```
// Port C initialization
// Func0=Out Func1=Out Func2=Out Func3=Out Func4=Out Func5=Out
Func6=Out Func7=Out
// State0=0 State1=0 State2=0 State3=0 State4=0 State5=0 State6=0 State7=0
```

```
PORTC=0x00;
DDRC=0xFF;
```

```
// Port D initialization
// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In
Func7=In
// State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T
```

```
PORTD=0x00;
```

```
DDRD=0x00;
```

```
// Timer/Counter 0 initialization  
// Clock source: System Clock  
// Clock value: Timer 0 Stopped  
// Mode: Normal top=FFh  
// OC0 output: Disconnected
```

```
TCCR0=0x00;  
TCNT0=0x00;  
OCR0=0x00;
```

```
// Timer/Counter 1 initialization  
// Clock source: System Clock  
// Clock value: Timer 1 Stopped  
// Mode: Normal top=FFFFh  
// OC1A output: Discon.  
// OC1B output: Discon.  
// Noise Canceler: Off  
// Input Capture on Falling Edge
```

```
TCCR1A=0x00;  
TCCR1B=0x00;  
TCNT1H=0x00;  
TCNT1L=0x00;  
OCR1AH=0x00;  
OCR1AL=0x00;  
OCR1BH=0x00;  
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization  
// Clock source: System Clock  
// Clock value: Timer 2 Stopped  
// Mode: Normal top=FFh  
// OC2 output: Disconnected
```

```
ASSR=0x00;  
TCCR2=0x00;  
TCNT2=0x00;  
OCR2=0x00;
```

```
// External Interrupt(s) initialization
```

```
// INT0: Off  
// INT1: Off  
// INT2: Off
```

```
MCUCR=0x00;  
MCUCSR=0x00;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization  
TIMSK=0x00;
```

```
// Analog Comparator initialization  
// Analog Comparator: Off  
// Analog Comparator Input Capture by Timer/Counter 1: Off  
// Analog Comparator Output: Off
```

```
ACSR=0x80;  
SFIOR=0x00;
```

```
PORTA=PING;  
TCNT0=0XB2;  
TCCR0=0X03;  
TIMSK=0X01;
```

```
While (1)  
{  
Scan();  
}  
}
```

## Listing program mikrokontroler 1

/\*\*\*\*\*\*

This program was produced by the  
CodeWizardAVR V1.24.2 full version  
Automatic Program Generator  
© Copyright 1998-2003 HP InfoTech s.r.l.  
<http://www.hpinfotech.ro>  
e-mail:office@hpinfotech.ro

Project : Robot Pemadam Api ( Fire Figthing Robot )  
Version : Full  
Date : 15/09/2008  
Author : Delta Agus S A  
Company : Delt@\_Ckp\_Software Development  
Jl Raya Kaligawe II / 2 No 22 Semarang  
Comments : trigger pulse ultrasonic range finder by delta\_master

Chip type : ATmega8535  
Program type : Application  
Clock frequency : 11.059200 MHz  
Memory model : Small  
External SRAM size : 0  
Data Stack size : 128  
\*\*\*\*\*/

```
#include <mega8535.h>
#include <delay.h>
#define PULSA PORTB.0
#define echo PINB.0
#define triger DDRB.0

#define PULSA_1 PORTB.2
#define echo_1 PINB.2
#define triger_1 DDRB.2

#define OUT 1
#define INPUT 0

unsigned int CACAH=0;
unsigned jarak;
unsingned floating;
```

```
void force(void)
{
    PORTC.2=1;
    PORTC.3=0;
}
```

```

void turbo(void)
{
    PORTC.2=0;
    PORTC.3=1;
}

void force_1(void)
{
    PORTC.4=1;
    PORTC.5=0;
}

void turbo_1(void)
{
    PORTC.4=0;
    PORTC.5=1;
}

void transmitter (void)
{

    while(echo==0){ };
    while(echo==1)
        PULSA=0;
        triger=INPUT;
        delay_us(5);

        triger=OUT;
        PULSA=1;
        delay_ms(5000);
    }
    cacah++{ };
    float(jarak);
    jarak=((cacah)/424*10)
    ;
    if (jarak <10){
        force();
    }
    else{
        turbo();
    }
}

void transmitter_2 (void)
{

    while(echo_1==0){ };
    while(echo_1==1)
        PULSA_1=0;
        triger_1=INPUT;
        delay_us(5);

```



```

        trigger_1=OUT;
        PULSA_1=1;
        delay_ms(5000);
    }
    cacah++{ };
    float(jarak);
    jarak=((cacah)/424*10);

    if (jarak <10){
        force_1();
    }
    else{
        turbo_1();
    }

void main (void)
{
    // Declare your local variables here
    // Input/Output Ports initialization
    // Port A initialization
    // Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In
    Func7=In
    // State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T

    PORTA=0xFF;
    DDRA=0x00;

    // Port B initialization
    // Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In
    Func7=In
    // State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T

    PORTB=0x00;
    DDRB=0x00;

    // Port C initialization
    // Func0=Out Func1=Out Func2=Out Func3=Out Func4=Out Func5=Out
    Func6=Out Func7=Out
    // State0=0 State1=0 State2=0 State3=0 State4=0 State5=0 State6=0 State7=0

    PORTC=0x00;
    DDRC=0xFF;

    // Port D initialization
    // Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In
    Func7=In
    // State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T

    PORTD=0x00;
    DDRD=0x00;

```

```
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
```

```
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
```

```
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
```

```
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
```

```
// External Interrupt(s) initialization
```

```
// INT0: Off
// INT1: Off
// INT2: Off
```

```
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
// Analog Comparator Output: Off

ACSR=0x80;
SFIOR=0x00;

PORTA=PING;
TCNT0=0XB2;
TCCR0=0X03;
TIMSK=0X01;

DDRC=0xFF;
while(1)
{
    transmitter();
    transmitter_1();
}
```

# LAMPIRAN



Data sheed